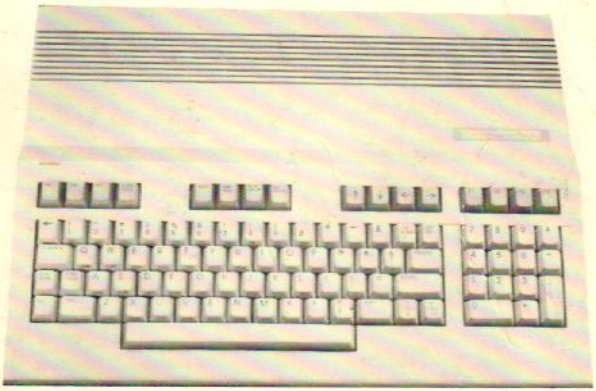
Commodore 128

PERSONAL COMPUTER

Commodore 128D

PERSONAL COMPUTER

CP/M-Modus
Kapitel 7



Commodore

7. CP/M-Modus

7.1 Einleitung
7.2 Starten von CP/M 3.0
7.3 CP/M Dateien
7.4 Kopieren der CP/M-Disketten und Dateien
7.5 Steuern der Ein- und Ausgabe
7.6 Arbeiten unter CP/M
7.7 Erweiterungen des CP/M 3.0 beim Commodore 128
7.8 CP/M 3.0 – Befehlsübersicht
7.9 CP/M 3.0 Transiente Dienstprogramme

Inhaltsverzeichnis

7. CP/M-Modus

7.1 Einleitung 7– 1
7.2 Starten von CP/M 3.0 ..... 7– 3
7.3 CP/M Dateien 7– 8
7.4 Kopieren der CP/M-Disketten und Dateien 7– 15
7.5 Steuern der Ein- und Ausgabe 7– 19
7.6 Arbeiten unter CP/M 7– 24
7.7 Erweiterungen des CP/M 3.0 ..... 7– 28
7.8 CP/M 3.0 – Befehlsübersicht 7– 34
7.9 CP/M 3.0 – Befehlsbeschreibungen 7– 39

Allgemeine Hinweise zu diesem Handbuch

Das Bildsignal des **80-Zeichen-Modus ist nur am RGBI-Ausgang** vorhanden, nicht am Video- bzw. Fernsehausgang, d.h. bei Anschluß eines Fernsehgerätes oder eines Monitors an der Fernsehbuchse bzw. an der Videobuchse kann nur im 40-Zeichen-Modus gearbeitet werden. Umgekehrt ist der 40-Zeichen-Modus nicht am RGBI-Ausgang verfügbar.

Die mitgelieferte CP/M-Systemdiskette ist beidseitig beschrieben. Wir empfehlen Ihnen von jeder Seite eine Arbeitskopie anzufertigen und die Originaldiskette an einem sicheren Ort aufzubewahren.

Alle Informationen, die in diesem Handbuch unter C 128 beschrieben sind gelten ebenfalls für den C 128 D.

7. CP/M–Modus

7.1 Einleitung

CP/M ist ein Produkt der Firma Digital Research. Die beim Commodore 128 verwendete CP/M Version ist

CP/M Plus Version 3.0.

In diesem Handbuch wird diese Version als CP/M 3.0 oder nur CP/M bezeichnet. Die Beschreibung bezieht sich hier auf die Anwendung des CP/M beim C128. Detailliertere Auskunft finden Sie im Handbuch für CP/M–Plus, das zusammen mit dem Uitility–Programm durch die Ihrer Garantieinformation anhängenden Bestellkarte angefordert werden kann.

CP/M 3.0 ist ein weit verbreitetes Betriebssystem für Mikrocomputer. CP/M 3.0 steuert und verwaltet alle Computerfunktionen: Speicher, Diskettenlaufwerke, Konsole (Bildschirm und Tastatur), Drucker und andere Peripheriegeräte. CP/M 3.0 verwaltet Diskettendateien und ermöglicht das Kopieren von Dateien in den Speicher oder zu anderen Peripheriegeräten wie z. B. zu einem Drucker. CP/M 3.0 legt verschiedene Programme im Computerspeicher ab und führt sie in Abhängigkeit von den Befehlen, die über die Konsole eingegeben werden, aus.

CP/M kann zum Erstellen eigener Programme oder zum Ausführen einer breiten Palette bereits vorhandener Programme verwendet werden.

7.1.1 Voraussetzungen zum Betrieb mit CP/M 3.0

Die notwendige Mindestkonfiguration zum Arbeiten unter dem Betriebssystem CP/M 3.0 besteht aus einem Computer mit einem 8080–, 8085– oder Z80–Mikroprozessor, einer Konsole (Tastatur und Bildschirm), einem Diskettenlaufwerk und einer Systemdiskette.

Im Commodore 128 ist der Z80–Mikroprozessor bereits fest eingebaut; als Konsole dienen die volle C128– Tastatur und ein 80–Zeichen Bildschirm; als Diskettenstation kann das neue schnelle Diskettenlaufwerk Commodore 1571 wie auch das Laufwerk 1541, dann allerdings nur mit normaler Leser/Schreib-Geschwindigkeit verwendet werden. Dazu gehören noch zwei Disketten, eine mit dem eigentlichen CP/M–Betriebssystem und einem ausführlichen HELP–Dienstprogramm und eine mit vielen anderen Dienstprogrammen.

7.1.2 Verwendung eines 40–Zeichen–Bildschirms

Bei der Verwendung eines 40–Zeichen–Bildschirms können nur 40 Zeichen pro Zeile gleichzeitig dargestellt werden. Um alle 80 Zeichen nacheinander zu sehen, muß der Bildschirminhalt horizontal verschoben werden. Dazu wird die CONTROL–Taste in Verbindung mit den entsprechenden Cursortasten (links oder rechts) verwendet.

7.1.3 Verwendung des Diskettenlaufwerks 1541

Auch dieses Laufwerk kann mit CP/M verwendet werden. Dabei erfolgt der Datentransfer zwischen der Konsole und Laufwerk wesentlich langsamer als mit dem Laufwerk Commodore 1571.

Das Diskettenlaufwerk 1541 kann, im Gegensatz zum Diskettenlaufwerk 1571, die Disketten nur einseitig bearbeiten. Dateien auf mit 1571 beschriebenen Disketten können nur solange bearbeitet werden, wie sie sich auf der ersten Diskettenseite befinden.

Mit beiden Laufwerken können auch Disketten, die mit Commodore 64 und seinem Z80–Modul angelegt sind, bearbeitet werden.

7.1.4 Verwendung anderer Diskettenformate

Um Dateien von anderen Computern bearbeiten zu können, ermöglicht das neue Laufwerk Commodore 1571 auch die Verwendung von Disketten, die mit folgenden Formaten beschrieben sind:

Epson QX10 Valdocs,
Kaypro II und
Osborne

unter dem Betriebssystem CP/M 80 und

IBM PC bzw. **Commodore PC10**

unter dem Betriebssystem CP/M 86.

Eine Erweiterung um andere Formate ist mit entsprechenden Programmen jederzeit möglich.

Wird eine mit einem fremden Format formatierte Diskette in das Laufwerk 1571 eingelegt, erscheint beim ersten Zugriff auf diese Diskette ein Leuchtbalken in der untersten Bildschirmzeile. In diesem Leuchtbalken erscheint dann die Systembezeichnung des Diskettenformats oder, wenn das Format nicht identifiziert werden kann, das Wort

'MISSING'.

Die vom Laufwerk akzeptierten Disketten können dann unter CP/M 3.0 beliebig gelesen und beschrieben werden. Damit stehen dem direkten Programm– und Datenaustausch mit anderen Systemen keine Hindernisse im Weg.

Nach der Meldung des Fremdformates, muß entweder Return zum einmaligen Zugriff, oder Control Return zum dauernden Zugriff gedrückt werden. Sollte es sich z. B. um eine Diskette Kaypro II handeln, so muß nach der Meldung (Kaypro II) mit der linken oberen Cursortaste Kaypro II angewählt werden.

Die Programmübernahme wird auch dadurch erleichtert, daß die C128–Konsole die Arbeitsweise des Terminals **ADM31** bzw. **ADM3A** emuliert. Die gleiche Emulation finden wir z. B. bei KAYPRO II und IV.

7.2 Starten des CP/M 3.0

7.2.1 Laden des CP/M 3.0

Unter dem Laden (boot) des CP/M ist ein Kopieren des Betriebssystems von der CP/M–Systemdiskette in den Speicher des Computers zu verstehen.

Dies kann auf verschiedene Arten geschehen. Bei ausgeschaltetem Computer kann man die CP/M–Systemdiskette in das Diskettenlaufwerk einlegen und dann den C128 einschalten. Das CP/M–Betriebssystem wird automatisch geladen. Wenn sich der C128 im BASIC–Modus befindet, können Sie den BASIC–Befehl BOOT zum Laden des CP/M verwenden oder die RESET–Taste drücken. Dabei muß sich die CP/M–Systemdiskette im Laufwerk befinden.

Achtung:
Wenn Sie eine Diskette in das Laufwerk einlegen, vergewissern Sie sich vor dem Verriegeln des Laufwerks, ob die Diskette bis zum Anschlag eingeschoben ist!

Wenn Sie aus dem C64– in den CP/M–Modus wechseln möchten, müssen Sie zuerst den Computer ausschalten und dann das CP/M wie oben beschrieben laden.

Unter CP/M 3.0 steht auf dem Commodore 128 ein freier Benutzerspeicher (TPA – Transient Program Area) von 59 KByte zur Verfügung.

7.2.2 Die CP/M–Startmeldung auf dem Bildschirm

Nachdem das CP/M in den Speicher geladen und gestartet wurde, erscheint folgende Meldung auf dem Bildschirm:

```
CP/M 3.0 On the Commodore 128
        xx column display

A>
```

Je nachdem, welcher Bildschirm angewählt ist, erscheint anstelle von `xx` die Zahl `40` oder `80`.

Der wichtigste Teil dieser Meldung sind die zwei folgenden Zeichen:

```
A>
```

Das ist die CP/M–System– bzw. Bereitschaftsmeldung (system prompt). Diese Meldung signalisiert die Bereitschaft von CP/M zu einer Befehlsübernahme von der Tastatur. Die Meldung sagt weiterhin aus, daß das Laufwerk A das sog. aktuelle Laufwerk ist. Das bedeutet, daß bis zu einer anderen Laufwerkseingabe von der Tastatur alle Diskettenoperationen auf dem Laufwerk A durchgeführt werden. Das Fehlen einer Nummernangabe vor dem A bedeutet, daß der Benutzerbereich 0 (USER 0) der aktuelle Benutzerbereich ist.

Unter CP/M wird ein Disketteneinzellaufwerk als Laufwerk A zugeordnet. Das entspricht der Geräteadresse 8, Laufwerk 0 im C128– und C64–Modus. Die maximale Anzahl der tatsächlich angeschlossenen Diskettenlaufwerke

beträgt vier. Weiteren Laufwerken sind die Laufwerksbezeichnungen B, C und D zugeordnet. Ein virtuelles Laufwerk, das auf dem Einzellaufwerk A ein weiteres Laufwerk emuliert, erhielt die Bezeichnung E.

7.2.3 Die CP/M Befehlszeile

CP/M 3.0 steuert die Aktivitäten des Computers in Abhängigkeit von den Befehlen, die Sie über die Tastatur eingeben. Diese Befehle erscheinen auf dem Bildschirm in einer **Befehlszeile**. Eine CP/M Befehlszeile setzt sich aus einem **Befehlsschlüsselwort** und aus optionellen **Befehlsargumenten** zusammen. Das Befehlsschlüsselwort bezeichnet entweder einen residenten CP/M–Befehl (z. B. TYPE, DIR u. a.) oder ein ausführbares Programm auf der Diskette. Das Argument beinhaltet zusätzliche Informationen, z. B. Dateiname oder andere Parameter. Nachfolgend ein Beispiel für eine Befehlszeile:

```
A>DIR TESTDATA
```

In diesem Beispiel ist DIR das Befehlsschlüsselwort und TESTDATA das Argument. Um die Befehlszeile an CP/M zu übergeben, muß die RETURN-Taste gedrückt werden. In dieser Anleitung wird das notwendige Drücken der RETURN–Taste mit **<CR>** dargestellt.

Beim Eingeben eines Zeichens auf der Tastatur erscheint dieses Zeichen gleichzeitig auch auf dem Bildschirm. Der Cursor bewegt sich dabei um eine Stelle nach rechts. Wenn Sie sich vertippen, drücken Sie die INST/DEL–Taste oder CTRL–H, um den Cursor nach links zurückzubewegen, und korrigieren Sie die falsche Eingabe. CTRL ist die Abkürzung für die CONTROL–Taste. Um ein Controlzeichen einzugeben, muß die CTRL–Taste gedrückt gehalten und gleichzeitig die entsprechende andere Taste betätigt werden. Manchmal wird für CONTROL auch die Abkürzung ↑ (Pfeil nach oben), z. B. ↑Z für CTRL–Z, verwendet.

Eine Übersicht der Control–Zeichen und deren Verwendung ist im Abschnitt 7.5 abgedruckt.

Alle Eingaben in der Befehlszeile können in kleinen bzw. großen Buchstaben erfolgen. CP/M interpretiert alle Zeichen in der Befehlszeile als großgeschrieben.

Die Eingabe in die Befehlszeile sollte unmittelbar nach der Bereitschaftsmeldung erfolgen, obwohl CP/M 3.0 auch Leerstellen zwischen der Bereitschaftsmeldung und Befehlswort toleriert.

7.2.4 Die Befehlsarten

CP/M 3.0 unterscheidet zwei verschiedene Befehlsarten: interne bzw. residente (built–in commands) und externe bzw. transiente Befehle (Dienst– und andere Programme).

Residente Befehle veranlassen die Ausführung von Programmen, die fest im Speicher als Teil des CP/M verankert sind. Diese Befehle können jederzeit ausgeführt werden, da dazu keine Diskettendateien notwendig sind.

Transiente Programme sind als Programmdateien auf Disketten gespeichert. Sie müssen zuerst von der Diskette geladen werden, bevor sie ausgeführt werden können. Diese Programmdateien sind durch die Typbezeichnung COM gekennzeichnet.

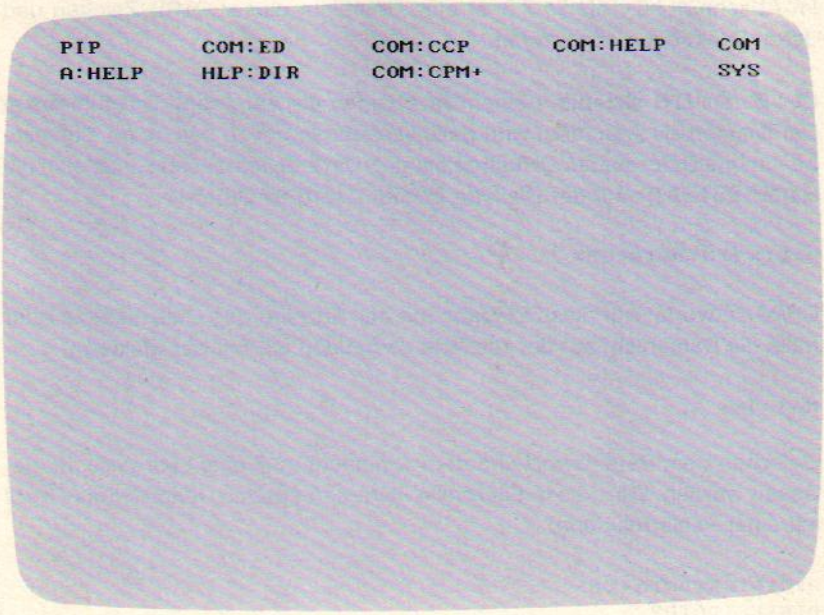
Bei den transienten Befehlen wird nur das Befehlsschlüsselwort von CP/M interpretiert und die entsprechende Datei von der Diskette geladen. Die Argumente werden durch CP/M nicht überprüft, sondern direkt für die Programme in einem Puffer abgelegt. Eine Befehlszeile darf nicht länger als 128 Zeichen sein.

7.2.5 Verarbeitung einer Befehlszeile durch CP/M

Nehmen wir den DIR–Befehl, um zu sehen, wie CP/M eine Befehlszeile interpretiert. DIR, das eine Abkürzung für 'directory' darstellt, veranlaßt CP/M, ein Verzeichnis der Diskettendateien auf dem Bildschirm anzuzeigen. Der Befehl DIR wird direkt nach der Systemmeldung eingegeben und mit RETURN abgeschlossen.

```
A>DIR<CR>
```

CP/M reagiert auf diesen Befehl mit der Anzeige der Namen aller Dateien, die auf der Diskette im Laufwerk A gespeichert sind. Wenn sich beispielsweise die CP/M– Systemdiskette im Laufwerk A befindet, wird folgendes auf dem Bildschirm angezeigt:



CP/M erkennt nur korrekt eingegebene Befehlsschlüsselworte. Wenn Sie bei der Eingabe einen Fehler machen und RETURN drücken, ohne den Fehler zu korrigieren, wiederholt CP/M die Befehlszeile und fügt ein Fragezeichen an. Wird z. B. fälschlich anstelle von DIR

```
A>DJR<CR>
```

eingegeben, reagiert CP/M wie folgt:

Auf der Diskette im Laufwerk A wird nach einer Datei DJR.COM gesucht. Ist diese Datei nicht vorhanden, wird auf dem Bildschirm

```
DJR?
```

angezeigt.

Dies bedeutet, daß der Befehl DJR vom CP/M nicht gefunden werden kann. Um einen Tippfehler auszubessern, können Sie die INST/DEL–Taste oder CTRL–H zum Löschen des letzten eingegebenen Zeichens verwenden.

CP/M stellt eine Reihe von CTRL–Zeichen zur Verfügung, die Ihnen beim Editieren der Befehlszeile helfen.

Im Abschnitt 7.6, Tabelle 7.3 ist eine Übersicht dieser CTRL–Zeichen und deren Funktionen abgedruckt.

Der Befehl DIR akzeptiert einen Dateinamen als Argument. DIR mit einem Dateinamen als Argument wird dazu verwendet, um zu sehen, ob sich eine bestimmte Datei auf der Diskette befindet. Um z. B. festzustellen, ob sich die Datei TESTDATA auf der Diskette befindet, wird eingeben:

```
A>DIRTESTDATA<CR>
```

CP/M antwortet auf diese Eingabe mit der Anzeige des Dateinamens oder (falls die Datei nicht auf der Diskette vorhanden ist) mit der Meldung:

```
No File
```

Zwischen dem Befehlswort und dem Argument muß eine Leerstelle ausgelassen werden. Wird diese Leerstelle nicht eingegeben, reagiert das CP/M mit einer Fehlermeldung:

```
A>DIRTESTDATA<CR>
DIRTESTDATA?
```

7.3 CP/M Dateien

Eine der wichtigsten CP/M–Eigenschaften ist das Verwalten und Bearbeiten von Dateien auf Disketten. Dateien im CP/M–Modus sind im Prinzip das gleiche wie im C128– oder C64–Modus – nämlich eine Datenmenge. Allerdings werden diese Dateien von CP/M anders als in den anderen Modi behandelt. Dieser Abschnitt macht Sie mit zwei Arten von Dateien bekannt; Sie werden lernen, wie man Dateien erstellt, benennt und bearbeitet und welche Dateien auf Ihren CP/M–Disketten abgespeichert sind.

Wie bereits gesagt, eine CP/M–Datei ist eine Menge von Daten. Jede Datei muß einen eindeutigen Namen haben, damit sie von CP/M erkannt wird. Auch ein Inhaltsverzeichnis (directory) ist auf jeder Diskette abgespeichert. In diesem Inhaltsverzeichnis sind die Namen aller Dateien und deren Lage auf der Diskette festgehalten.

7.3.1 Zwei Arten von CP/M–Dateien:

Programmdateien und **Datendateien.**

In einer **Programmdatei** sind ausführbare Befehle abgespeichert, die vom Computer schrittweise abgearbeitet werden können. Eine **Datendatei** beinhaltet verschiedene Informationen, z. B. ein Namens– oder ein Adressverzeichnis oder einen Text. Die Datendateien können selbst nicht ausgeführt, sondern nur von den Programmdateien verarbeitet werden.

7.3.2 Erstellen einer Datei

Es gibt verschiedene Möglichkeiten, um CP/M–Dateien zu erstellen. Beispielsweise könnte der CP/M–Texteditor ED dazu verwendet werden, um Dateien zu erstellen und zu benennen. Sie können auch eine Datei erstellen, indem Sie eine bereits existierende Datei an eine andere Stelle kopieren; dabei kann auch der Name der Datei geändert werden. Unter CP/M kann dazu das Programm PIP verwendet werden. Auch andere Programme, z. B. MAC, der CP/M 3.0–Assembler, erzeugen Ausgabedateien, und diese können wiederum als Eingabedateien für andere Programme verwendet werden.

Die ED– und PIP–Befehle sind, zusammen mit anderen CP/M–Befehlen, ausführlich im Teil 7.9 aufgeführt.

7.3.3 Die Dateibezeichnung

CP/M erkennt jede Datei an ihrer eindeutigen Bezeichnung. Für die Dateibezeichnung werden folgende Teile verwendet:

- die Laufwerksbezeichnung
- der Dateiname
- die Typbezeichnung
- das Paßwort

Unbedingt erforderlich ist die Angabe des Dateinamens; alle anderen Bestandteile der Dateibezeichnung können, müssen aber nicht unbedingt angegeben werden.

7.3.4 Die Laufwerksbezeichnung

Die Laufwerksbezeichnung besteht aus einem Buchstaben (A – E) mit einem nachfolgenden Doppelpunkt. Jedem CP/M– Diskettenlaufwerk ist ein Buchstabe zugeordnet. Wenn sie die Laufwerksbezeichnung als Teil der Dateibezeichnung angeben, sucht CP/M auf diesem Laufwerk nach der angegebenen Datei.

Wenn Sie z. B.:

```
A>B:TESTDATA<CR>
```

eingeben, wird auf dem Diskettenlaufwerk B nach der Datei TESTDATA gesucht. Ohne Angabe der Laufwerksbezeichnung wird auf dem aktuellen Laufwerk (in der Regel A) nach der Datei gesucht.

7.3.5 Der Dateiname

Ein Dateiname kann bis zu 8 Zeichen lang sein. Eine Dateibezeichnung kann im einfachsten Fall nur aus einem Dateinamen bestehen.

Der Dateiname sollte so gewählt werden, daß man aus ihm auf den Inhalt der Datei schließen kann. Wenn Sie beispielsweise ein Kundenverzeichnis erstellen, können Sie als Namen dieser Datei

```
KUNDEN
```

wählen, so daß Sie schon an dem Dateinamen den Inhalt der Datei erkennen.

7.3.6 Die Typbezeichnung

Um Dateien, die ähnliche Strukturen aufweisen, auch bei unterschiedlichen Dateinamen zusammenfassen zu können, läßt sich wahlweise eine Typbezeichnung an den Dateinamen anhängen. Diese Typbezeichnung kann bis zu 3 Zeichen lang sein und wird vom eigentlichen Dateinamen mit einem Punkt (.) abgetrennt. Auch bei der Wahl dieser Typbezeichnung sollte die Bezeichnung in einem Zusammenhang mit der Art der Datei stehen.

Unser Kundennamensverzeichnis könnte beispielsweise diese Bezeichnung bekommen:

```
KUNDEN.NAM
```

Bei der Anzeige von Dateibezeichnungen durch CP/M werden kurze Dateinamen mit Leerzeichen auf acht Stellen aufgefüllt, so daß ein Vergleich der Typbezeichnungen schnell erfolgen kann.

Die **Programmdateien**, die vom CP/M in den Speicher geladen und ausgeführt werden, sind immer durch die Typbezeichnung COM gekennzeichnet.

7.3.7 Das Paßwort

Bei CP/M 3.0 auf dem Commodore 128 kann ein Paßwort als Teil der Dateibezeichnung definiert werden. Das Paßwort kann bis zu 8 Zeichen lang sein und wird bei der Eingabe mit einem Semikolon von der vorangestellten Dateibebezeichnung getrennt:

```
KUNDEN.NAM;LISTE
```

Das Paßwort kann wahlweise eingegeben werden. Allerdings **muß** dieses Paßwort als Teil der Dateibezeichnung immer eingegeben werden, wenn der Zugriff auf eine Datei mit einem Paßwort geschützt ist.

Beispiel einer vollständigen Dateibezeichnung:

```
A:DOKUMENT.GES;EDGAR
```

7.3.8 Dateispezifikation 'dateispez'

Als 'dateispez' (Dateispezifikation) wird jede zulässige Kombination der 4 Teile der Dateibezeichnung abgekürzt. Alle Teile müssen mit ihren passenden Trennzeichen getrennt werden.

Zulässige Kombinationen für CP/M–Dateibezeichnungen sind:

Dateiname
Laufwerksbezeichnung:Dateiname
Dateiname.Typbezeichnung
Laufwerksbezeichnung:Dateiname.Typbezeichnung

Dateiname;Passwort
Laufwerksbezeichnung:Dateiname;Passwort
Dateiname.Typbezeichnung;Passwort
Laufwerksbezeichnung:Dateiname.Typbez;Paßwort

7.3.9 Benutzerbereich (User)

CP/M kann die Dateien auch einem Benutzerbereich (User 0 – 15) zuordnen. Mit den Benutzerbereichsnummern (User–Nummern) können Dateien in 16 verschiedene Bereiche eingeteilt werden. Die Benutzerbereichsnummer wird der Datei bei deren Erstellung zugeordnet.

Die Benutzerbereichsnummer wird immer der CP/M–Bereitschaftsmeldung vorangestellt (mit Ausnahme der Nummer 0). Der Benutzerbereich 0 ist der voreingestellte Bereich, seine Nummer wird in der Bereitschaftsmeldung nicht angezeigt. Nachfolgend einige Beispiele für die Anzeige der Benutzerbereiche und deren Bedeutung:

`4A>` Benutzerbereich 4, Laufwerk A
`A>` Benutzerbereich 0, Laufwerk A
`2B>` Benutzerbereich 2, Laufwerk B

Zum Ändern des aktuellen Benutzerbereichs kann der interne Befehl USER verwendet werden:

```
A>USER 3<CR>
3A>
```

Sie können auch gleichzeitig den Benutzerbereich und das aktuelle Laufwerk ändern, indem Sie beides gleichzeitig eingeben:

```
A>3B:<CR>
3B>
```

Die meisten Befehle haben nur auf die Dateien Zugriff, die dem aktuellen Benutzerbereich zugeordnet sind. Allerdings kann man aus allen Benutzerbereichen auf die Dateien zugreifen, die dem Benutzerbereich 0 zugeordnet und mit dem SYS–Attribut gekennzeichnet sind.

7.3.10 Platzhalter beim Zugriff auf mehrere Dateien

Jeder interne oder transiente CP/M–Befehl kann sich auf mehrere Dateien beziehen, wenn im Dateinamen oder der Typbezeichnung besondere Platzhalterzeichen verwendet werden.

Ein Platzhalter ist ein Zeichen, das anstelle aller anderen zulässigen Zeichen verwendet werden kann. CP/M verwendet den Stern (*) und das Fragezeichen (?) als Platzhalter. Wird z. B. das Fragezeichen (?) als drittes Zeichen in einem Dateinamen eingesetzt, wird von CP/M an dieser Stelle jedes beliebige Zeichen akzeptiert. Ein * veranlaßt CP/M, den Dateinamen oder die Typbezeichnung von der Stelle, an der er eingesetzt wird, mit ? aufzufüllen. Mit einer Dateibezeichnung, die Platzhalter beinhaltet, kann CP/M entsprechend den Vorgaben auf mehrere Dateien zugreifen. Wenn z. B. folgendes eingegeben wird:

```
?????TAX.LIB
```

ordnet CP/M alle Dateien, die im Dateinamen auf TAX enden und die Typbezeichnung LIB haben, dieser Dateibezeichnung zu. Anstelle der ersten fünf Zeichen können beliebige Zeichen stehen.

7.3.11 Reservierte Zeichen

Die Zeichen, die in der folgenden Tabelle 7.1 aufgeführt sind, haben unter CP/M eine besondere Bedeutung und dürfen nicht als Bestandteil der Dateibezeichnung verwendet werden.

Tabelle 7.1 CP/M 3.0: Reservierte Zeichen

Zeichen	Bedeutung
< = , ! | > [ ] TAB CR LEERST.	Befehlszeile – Trennzeichen
:	Laufwerksbezeichnung – Trennzeichen
.	Typbezeichnung – Trennzeichen
;	Paßwort – Trennzeichen oder Kommentar– Trennzeichen am Anfang der Befehlszeile
*?	Platzhalter
[ ]	Befehlsargumente – Trennzeichen für globale und lokale Optionen
( )	Trennzeichen für mehrfache Angaben innerhalb von ekkigen Klammern
/$	Weitere Trennzeichen in der Befehlszeile
< > & ! | \ + –	Weitere Trennzeichen

7.3.12 Reservierte Typbezeichnungen

Unter CP/M 3.0 sind mehrere Dateitypen bestimmten Programmen bereits fest zugeordnet. In der nachfolgenden Tabelle 7.2 werden sie aufgelistet und kurz beschrieben.

Tabelle 7.2 CP/M 3.0: Reservierte Typbezeichnungen

Typ	Bedeutung
ASM	Quelldatei für Assembler ASM
BAS	BASIC Programm
COM	Ausführbare Programmdatei
HEX	Ausgabedatei von MAC (für HEXCOM)
HLP	HELP–Datei
$$$	Zwischendatei (temporär)
PRN	Printdatei von MAC oder RMAC
REL	Ausgabedatei von RMAC (für LINK)
SUB	Befehlsliste für SUBMIT
SYM	Symboldatei von MAC, RMAC oder LINK

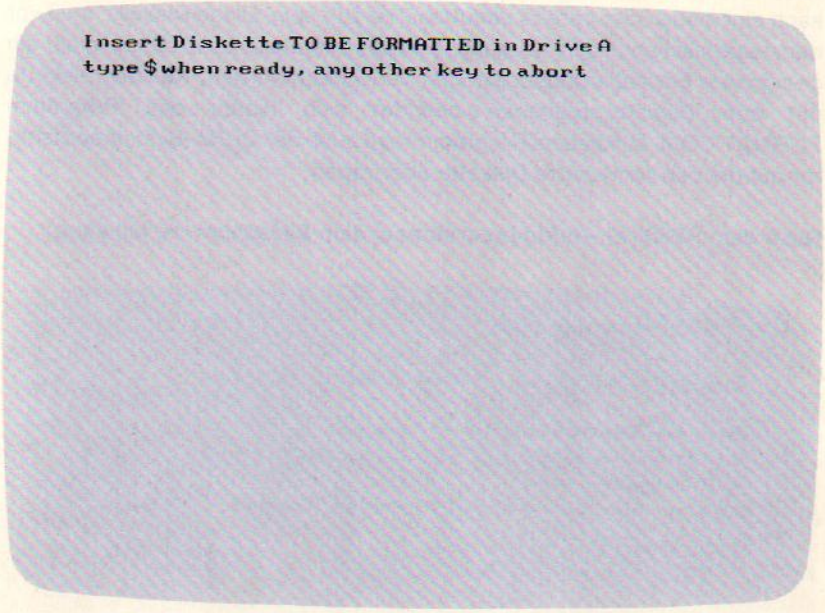
7.4 Kopieren der CP/M–Disketten und –Dateien

Kopien (backups) der CP/M–Disketten können mit einem oder zwei Diskettenlaufwerken angefertigt werden. Die Diskette, auf welche übertragen wird, kann eine neue oder bereits benutzte Diskette sein. Diese Diskette muß vor dem eigentlichen Kopiervorgang bereits formatiert sein. Dazu muß ein geeignetes Formatierungsprogramm verwendet werden. Auf der Diskette mit den Dienstprogrammen befindet sich hierzu das Programm FORMAT.COM. Mit diesem Programm wird auch der CP/M–Systembootsektor auf die neu formatierte Diskette übertragen.

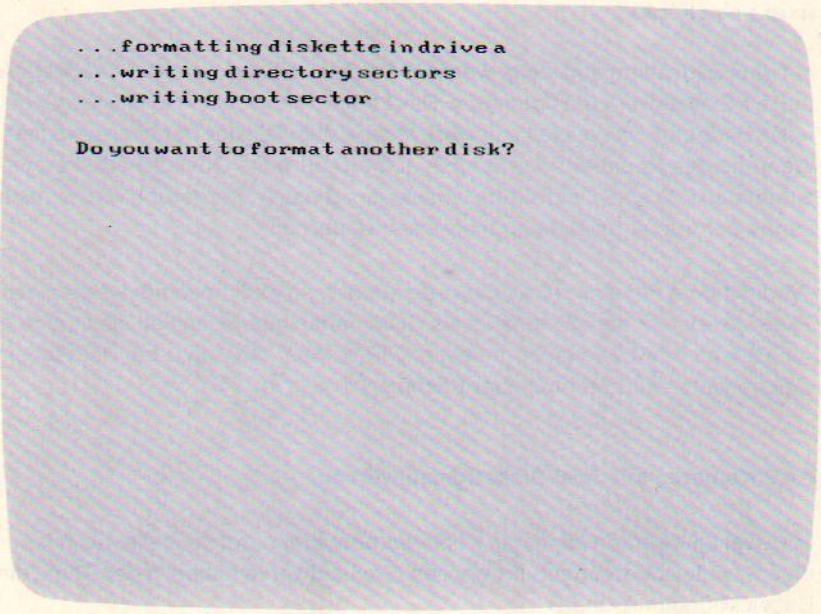
Nach dem FORMAT–Aufruf erscheint auf dem Bildschirm die Meldung:

```
C128 FORMAT PROGRAM

Drive A is a 1571       ; bzw. 1541

Please select disk type to format
C128 double sided       ; nicht mit 1541
C128 single sided
C64  single sided       ; Z80-Modul-Format
```

Mit den Cursortasten wird ein inverser Zeiger in die Zeile mit der gewünschten Formatbezeichnung bewegt und die RETURN–Taste gedrückt. Dann erscheint:



Nach dem Einlegen der zu formatierenden Diskette in Laufwerk A muß die $–Taste gedrückt werden. Es erscheint:



Beim Eingeben von N wird mit der Meldung

```
. . . exiting format program
```

das FORMAT–Programm beendet.

Alle anderen Dateien, auch die Systemdateien CPM+.SYS und CCP.COM, können mit dem Dienstprogramm PIP kopiert werden.

7.4.1 Kopieren mit einem Diskettenlaufwerk

Disketteninhalte können von einer auf eine andere Diskette auch mit nur einem Diskettenlaufwerk (1541 oder 1571) übertragen werden. Dazu wird

das Dienstprogramm PIP verwendet. Das Format für die PIP–Befehlseingaben kann mit dem HELP–Befehl:

```
A>HELP PIP<CR>
```

auf dem Bildschirm angezeigt werden. Als Quellaufwerk wird das Laufwerk A und als Ziellaufwerk E eingegeben. Das Laufwerk E ist ein virtuelles Laufwerk – das bedeutet, daß dieses Laufwerk nicht als ein Teil der Hardware existiert. Während des Kopierens werden Sie aufgefordert, die Quelldiskette herauszunehmen und eine Zieldiskette einzulegen. Diese Aufforderung wird je nach Länge der kopierten Dateien mehrmals wiederholt.

Wird mit dem PIP eine ganze Diskette kopiert, so muß beachtet werden, daß Files, die eine andere Extension als COM haben, auf jeden Fall die Option O für Objectcode bekommen müssen: PIP b:=a*.*ÄoÜ (ä,ü bei deutschem Zeichensatz, statt rechteckiger Klammer).

7.4.2 Kopieren mit zwei Diskettenlaufwerken.

Mit zwei Laufwerken kann das Kopieren bequemer und schneller als mit nur einem Laufwerk durchgeführt werden. Die Laufwerke haben in der Regel die Bezeichnungen A und B.

Um die CP/M–Systemdateien zu kopieren, wird das PIP–Programm aufgerufen:

```
A>PIP<CR>
CP/M 3 PIP VERSION 3.0
*
```

Jetzt wird z. B. eingegeben:

```
*b:=c*.*
```

und während des Kopierens erscheint nacheinander:

```
COPYING -
CPM+.SYS
CCP.COM
*
```

Anschließend können noch andere Dateien übertragen oder das PIP–Programm mit CTRL–C beendet werden.

7.5 Steuern der Ein– und Ausgabe

7.5.1 Steuern der Konsolenausgabe.

Die Anzeige von Informationen auf dem Bildschirm erfolgt oft so schnell, daß nicht mitgelesen werden kann. Um dem System mitzuteilen, daß die Ausgabe angehalten werden soll, drücken Sie die CTRL–Taste und gleichzeitig die Taste S. Das CTRL–S–Zeichen bewirkt ein Anhalten der Ausgabe. Die Ausgabe wird fortgesetzt, indem CTRL–Q gedrückt wird. Auch das Drücken der NO SCROLL–Taste hält die Ausgabe an. In der untersten Zeile (25) erscheint invers eine PAUSE–Anzeige. Bei wiederholtem Drücken der NO SCROLL–Taste wird die Ausgabe wieder fortgesetzt. Wird während der Pause eine andere Taste als CTRL–Q oder NO SCROLL betätigt, ertönt ein Piepton.

Einige CP/M–Programme (z. B. DIR oder TYPE) verwenden automatische seitenweise Ausgabe (paging) auf der Konsole. Das bedeutet, daß bei einer längeren Ausgabe, als ein Bildschirminhalt fassen kann, die Ausgabe bei gefülltem Bildschirm selbständig angehalten wird. Wenn dies geschieht, erscheint in der untersten Zeile die Aufforderung:

```
Press RETURN to continue
```

Erst dann wird die Ausgabe wieder fortgesetzt. Diese Option kann mit dem SETDEF–Befehl ein– oder ausgeschaltet werden.

7.5.2 Steuern der Ausgabe auf dem Drucker

Mit einem CTRL–Zeichen kann die Konsolenausgabe auch parallel auf einen Drucker geleitet werden (printer echo). Um die gleichzeitige Ausgabe auf den Drucker zu starten, wird CTRL–P eingegeben. Dabei ertönt ein Piepton.

Beendet wird die Druckerausgabe durch erneuerte Eingabe von CTRL–P. Bei eingeschalteter Druckerausgabe wird jedes Zeichen, das auf dem Bildschirm erscheint, auch auf dem Drucker ausgedruckt. Bei nichtbereitem Drucker wird auch die Bildschirmausgabe solange blockiert, bis der Drucker wieder bereit wird.

Zum Erstellen einer Liste der Dateien, die sich auf einer Diskette befinden, kann der Befehl DIR mit nachfolgender Eingabe von CTRL–P (vor dem RETURN) verwendet werden.

Auch bei der Druckerausgabe können die Zeichen CTRL–S, CTRL–Q und die Taste NO SCROLL verwendet werden, um die Ausgabe zu steuern. Soll ein Teil nicht mitgedruckt werden, wird die Ausgabe mit CTRL–S gestoppt, die Druckerausgabe mit CTRL–P abgeschaltet und die Bildschirmausgabe mit CTRL–Q fortgesetzt. Dabei können die Zeichen CTRL–P, CTRL–S und CTRL–Q beliebig kombiniert werden.

7.5.3 Zeileneditierung auf der Konsole

Wie bereits erwähnt, können Sie einfache Tippfehler mit den Tasten INST/DEL bzw. CTRL–H korrigieren. CP/M stellt außerdem noch andere Editierfunktionen in Form von CTRL–Zeichen zur Verfügung. Mit diesen CTRL–Zeichen können CP/M–Befehlszeilen oder Eingabezeilen anderer Programme editiert werden.

7.5.4 CTRL–Zeichen zum Editieren

Bei der Verwendung von CTRL–Zeichen, die in Tabelle 7.3 aufgeführt sind, können Sie den Cursor nach links und rechts zum Einfügen oder Löschen von Zeichen in der Mitte der editierten Zeile bewegen. Das bedeutet, daß Sie die Zeichen, die rechts von der korrigierten Stelle stehen, nicht neu eintippen müssen.

Im folgenden Beispiel wird gezeigt, wie eine falsche Eingabe von PIP korrigiert werden kann:

```
A>POP A:=B:*.*<CR>      (PIP falsch eingegeben)
?POP
A>POP A:=B:*.*          (CTRL–W wiederholt die Befehlszeile)
A>POP A:=B:*.*          (CTRL–B bewegt den Cursor an den
                         Zeilenanfang)
A>P110P A:=B:*.*        (CTRL–F bewegt den Cursor nach rechts)
A>PP A:=B:*.*           (CTRL–G löscht Fehler)
A>PI1P0 A:=B:*.*        (das I wird eingegeben)
```

Nach der Korrektur kann die RETURN–Taste gedrückt werden, auch wenn der Cursor noch in der Mitte der Zeile plaziert ist. Der RETURN–Befehl veranlaßt nicht nur die Ausführung der Befehlszeile, sondern speichert die Zeile auch in einen Puffer, so daß sie mit CTRL–W wieder aufgerufen werden kann.

Wenn Sie ein Zeichen in die Mitte der Befehlszeile einfügen, bewegen sich alle Zeichen rechts vom Cursor weiter nach rechts. Wird die Zeile länger als die Bildschirmbreite, verschwinden die Zeichen auf der rechten Bildschirmseite aus dem Bild. Diese Zeichen sind aber nicht gelöscht. Sie werden wieder angezeigt, wenn Sie Zeichen in der Zeile löschen oder CTRL–E drücken, sobald sich der Cursor in der Mitte der Zeile befindet. Mit CTRL–E werden alle Zeichen rechts vom Cursor in die nächste Bildschirmzeile bewegt.

Tabelle 7.3 CP/M 3.0: Editier– und Steuer–CTRL–Zeichen

Zeichen	Bedeutung
CTRL–A	Bewegt den Cursor eine Stelle nach links.
CTRL–B	Bewegt den Cursor an den Anfang der Befehlszeile, ohne in der Zeile eine Veränderung hervorzurufen. Wenn sich der Cursor am Anfang der Zeile befindet, wird er mit CTRL–B auf das Zeilenende bewegt, usw.
CTRL–C	Unterbricht den Programmablauf und setzt die Laufwerksparameter zurück.
CTRL–E	Bewirkt einen Zeilenvorschub ohne den Befehl an CP/M zu übergeben. Bewegt den Cursor an den Anfang der nächsten Zeile, ohne die bisherigen Eingaben zu löschen.
CTRL–F	Bewegt den Cursor um eine Stelle nach rechts.
CTRL–G	Löscht das Zeichen unter dem Cursor. Der Cursor wird dabei nicht bewegt. Die Zeichen rechts vom Cursor werden um eine Stelle nach links bewegt.
CTRL–H	Löscht das Zeichen links vom Cursor und bewegt den Cursor um eine Stelle nach links. Die Zeichen rechts vom Cursor werden um eine Stelle nach links verschoben.

Tabelle 7.3 CP/M 3.0: Editier– und Steuer–CTRL–Zeichen (Fortsetzung)

Zeichen	Bedeutung
CTRL–I	Bewegt den Cursor zur nächsten TAB–Marke. Die TAB-Marken sind automatisch auf jede achte Stelle gesetzt. Hat die gleiche Funktion wie die TAB–Taste.
CTRL–J	Übergibt die Befehlszeile an CP/M und setzt den Cursor an den Anfang der nächsten Zeile. Hat die gleiche Funktion wie die RETURN–Taste oder CTRL–M.
CTRL–K	Löscht alle Zeichen rechts vom Cursor.
CTRL–M	Übergibt die Befehlszeile an CP/M und setzt den Cursor an den Anfang der nächsten Zeile. Hat die gleiche Funktion wie die RETURN–Taste oder CTRL–J.
CTRL–P	Schaltet parallel zur Bildschirmausgabe auch die Drukkerausgabe ein.
CTRL–Q	Die Bildschirmausgabe wird fortgesetzt.
CTRL–R	Wiederholt die Befehlszeile bis zum Cursor in der nächsten Zeile. An der Cursorposition erscheint ein #–Zeichen, und der Cursor wird um eine Zeile in derselben Spalte nach unten bewegt.
CTRL–S	Die Bildschirmausgabe wird so lange angehalten, bis CTRL–Q eingegeben wird.
CTRL–U	Löscht alle Zeichen in der Befehlszeile, setzt ein #–Zeichen an die Cursor–Position und bewegt den Cursor in die nächste Zeile. Mit CTRL–W können alle Zeichen, die links neben dem Cursor in der alten Zeile gestanden haben, in der neuen Zeile wiederholt werden.

Tabelle 7.3 CP/M 3.0: Editier– und Steuer–CTRL–Zeichen Fortsetzung)

Zeichen	Bedeutung
CTRL–W	Wiederholt die letzte ausgeführte Befehlszeile, ohne sie an CP/M zu übergeben. Der Cursor muß dabei am Anfang der Befehlszeile stehen, sonst wird CTRL–W nicht ausgeführt. Wiederholt wird die letzte mit CTRL–J, CTRL–M, CTRL–U oder RETURN abgeschlossene Zeile. Wenn sich in der Befehlszeile bereits Zeichen befinden, wird mit CTRL–W der Cursor lediglich an das Ende der Zeile bewegt. Mit <CR> wird die wiederholte Zeile an CP/M übergeben.
CTRL–X	Löscht alle Zeichen links vom Cursor und bewegt den Cursor an den Anfang der Zeile. Alle Zeichen rechts vom Cursor werden ab Zeilenanfang kopiert.
CTRL–Z	Dateiendezeichen.

7.6 Arbeiten unter CP/M

7.6.1 Umleitung von Eingaben und Ausgaben

Der Befehl PUT ermöglicht es, die Ausgabe auf den Bildschirm oder auf den Drucker in eine Diskettendatei umzuleiten. Der GET–Befehl wird dazu verwendet, die Konsoleneingabe für CP/M oder ein Programm anstelle von der Konsole aus einer Diskettendatei vorzunehmen. Die nachfolgenden Beispiele zeigen einige der Möglichkeiten, die sich hinter GET und PUT verbergen.

PUT wird verwendet, um die Ausgabe von der Konsole in eine Diskettendatei so umzuleiten, als ob es die Konsole wäre. Mit PUT kann z. B. eine Diskettendatei erstellt werden, in der das Inhaltsverzeichnis der Diskette so gespeichert wird, wie es sonst auf dem Bildschirm ausgegeben wird.

```
A>PUT CONSOLE OUTPUT TO FILE DIR.PRN <CR>
PUTTING CONSOLE OUTPUT TO FILE DIR:PRN

A>DIR<CR>
A>NAME     TEX:FRONT    TEX:FRONT    BAK:ONE      TEX
A>FOUR     TEX:ONE      BAK:LINEDIT  TEX:EXAMP1   TXT
A>TWO      TEX:THREE    BAK:EXAMP2                TXT

A>TYPE DIR.PRN<CR>
A>NAME     TEX:FRONT    TEX:FRONT    BAK:ONE      TEX
A>FOUR     TEX:ONE      BAK:LINEDIT  TEX:EXAMP1   TXT
A>TWO      TEX:THREE    BAK:EXAMP2   TXT
```

Der GET–Befehl ermöglicht die Verarbeitung von Daten, die normalerweise über die Tastatur eingegeben werden, aus einer Diskettendatei. Wird diese Datei von CP/M gelesen, muß sie aus normalen CP/M–Befehlszeilen zusammengeetzt sein.

Wenn die Diskettendatei einem Programm als Eingabe dienen soll, muß sie sich aus den vom jeweiligen Programm lesbaren Befehlen und Daten zusammensetzen. Die Eingabedatei darf sowohl CP/M–Befehle als auch Programmeingaben beinhalten.

7.6.2 Zuordnung logischer Einheiten

Die minimale Commodore–128–CP/M 3.0–Hardware–Konfiguration besteht aus Rechner mit Tastatur, Bildschirm und einem Diskettenlaufwerk. Sie können auch andere Einheiten an ihr System anschließen, z. B. einen Drucker oder ein Modem. Die Zuordnung der logischen und physikalischen Einheiten zu den Einheitennamen ist aus der Tabelle 7.4 ersichtlich. Sie zeigt gleichzeitig auch die Zuordnung der physikalischen zu den logischen Einheiten beim Commodore 128 an.

Mit dem DEVICE–Befehl kann die Zuordnung geändert werden. AUXIN und AUXOUT können beispielsweise einer seriellen Schnittstelle (6551) zugeordnet werden, an die ein Modem angeschlossen werden kann.

Tabelle 7.4 CP/M 3.0: Logische Einheiten

Logischer Name der Einheit	Einheiten Typ	Physikalische Zuordnung der Einheit
CONIN:	Konsolen–eingabe	KEYS (Tastatur)
CONOUT:	Konsolen–ausgabe	80COL oder 40COL (Bildschirm)
AUXIN:	Externe Eingabe	—
AUXOUT:	Externe Ausgabe	—
LST:	List Ausgabe	PTR1 oder PTR2 (Drucker)

Bei CP/M auf dem C128 ist eine physikalische Einheit 6551 vorhanden. Diese Einheit emuliert ein ACIA–IC und kann als serielle Schnittstelle verwendet werden.

Durch entsprechende Zuordnung mit dem DEVICE–Befehl kann diese Einheit als Modem– oder Druckeranschluß dienen.

7.6.3 Suchen von Programmdateien

Wenn ein nicht–residenter Befehl eingegeben wird, sucht CP/M auf dem aktuellen oder angegebenen Laufwerk nach einer Datei mit diesem Namen und dem Dateityp COM. Hierbei wird zuerst im aktuellen Benutzerbereich gesucht und dann im Benutzerbereich 0 nach einer gleichnamigen Datei mit dem SYS–Attribut. Sobald die entsprechende Programmdatei gefunden worden ist, wird der Inhalt dieser Datei in den Speicher (TPA) geladen und das Programm ausgeführt. Nach Ablauf des Programms meldet sich CP/M wieder mit der Bereitschaftsmeldung und erwartet die Eingabe eines weiteren Befehls. Wurde die angegebene Datei nicht gefunden, wiederholt CP/M die Befehlszeile mit einem angehängten ? und wartet ebenfalls auf die nächste Befehlseingabe.

7.6.4 Ausführen von Mehrfachbefehlen

In den bisherigen Beispielen hat CP/M immer nur einen Befehl ausgeführt. CP/M kann aber auch eine Befehlsfolge abarbeiten. Die Befehlsfolge kann entweder in der Befehlszeile eingegeben oder als eine Diskettendatei mit der Typbezeichnung SUB auf einer Diskette abgespeichert werden. Die in einer SUB–Diskettendatei abgespeicherte Befehlsfolge kann mit dem SUBMIT-Befehl abgearbeitet werden.

7.6.5 Unterbrechen des Programmablaufs

Das Zeichen CTRL–C kann zum Unterbrechen des Programmablaufs oder zum Rücksetzen der Diskettenlaufwerke verwendet werden.

Viele Programme lassen sich in ihrem Ablauf mit der Eingabe von CTRL–C unterbrechen. Nur wenn versucht wird, einen Programmablauf während der Datenausgabe auf den Bildschirm zu unterbrechen, muß zuerst die Ausgabe mit CTRL–S angehalten und erst dann CTRL–C eingegeben werden.

7.6.6 HELP–Programm

Mit dem transienten Befehl HELP kann man aus einer Hilfsdatei Informationen über die meisten CP/M–Befehle, deren Eingaben und deren Bedienung auf den Bildschirm bringen. Um das Programm aufzurufen, genügt es,

```
HELP<CR>
```

einzugeben. Sie können auch die HELP–Taste drücken und anschließend RETURN betätigen.

Auf dem Bildschirm erscheint ein Verzeichnis der abrufbaren Informationen:

```
Topics available:

COMMANDS   CNTRCHARS  COPYSYS  DATE      DEVICE   DIR
DUMP       ED         ERASE    FILESPEC  GENCOM   GET
PATCH      PIP(COPY)  PUT      RENAME    RMAC     SAVE
SET        SETDEF     SHOW     SID       SUBMIT   TYPE
USER       XREF
```

Um beispielsweise Informationen zum Programm PIP zu bekommen, geben Sie ein:

```
HELP> PIP<CR>
```

Folgendes wird angezeigt:

```
PIP (COPY)

Syntax:
                DESTINATION=SOURCE

PIP d:{Gn}|filespec{[Gn]} = filespec{[o]}, . . .| d:{[o]}

Explanation:

The file copy program PIP copies files, combines files, and
transfers files between disks, printers, consoles, or other
devices attached to your computer. The first filespec is the
destination. The second filespec is the source. Use two or
more source filespecs separated by commas to combine two or
more files into one file. [o] is any combination of the avai-
lable options. The ;Gn? option in the destination filespec
tells PIP to copy your file to that user number.

PIP with no command tail displays an * prompt and awaits your
series of commands, entered and processed one line at a time.
The source or destination can be any CP/M 3.0 logical device.
```

Mit HELP kann man schnell benötigte Informationen über die internen und transienten CP/M–Befehle abrufen. Wenn Sie Informationen gezielt abrufen wollen, können Sie den Programmnamen in der Befehlszeile eingeben, und die benötigte Information wird sofort angezeigt:

```
A>HELP PIP<CR>
A>HELP DIRSYS<CR>
```

Sie können mit dem HELP–Befehl auch die CP/M–Befehlsbeschreibungen durchblättern, um ihre Kenntnisse über CP/M aufzufrischen und zu vertiefen.

7.7 Erweiterungen des CP/M 3.0 beim Commodore 128

Beim CP/M 3.0 auf dem Commodore 128 sind dem Betriebssystem einige Erweiterungen zugefügt, die sonst in CP/M nicht enthalten sind. Diese Erweiterungen werden hier im folgenden beschrieben.

7.7.1 Tastaturerweiterungen

Jeder Taste kann ein anderer Code bzw. andere Bedeutung als ursprünglich zugeordnet werden. Hiervon sind jedoch folgende Tasten ausgenommen:

Linke SHIFT– Taste
Rechte SHIFT– Taste
COMMODORE– Taste

CONTROL– Taste
RESTORE– Taste
40/80– Taste
ASCII/DIN– Taste

Um die Belegung einer Taste zu definieren, gibt es bei CP/M zusätzliche Tastaturfunktionen. Um diese Funktionen aufzurufen, müssen mehrere Tasten gleichzeitig gedrückt werden – zuerst die CTRL– und die rechte SHIFT–Taste und dann nach Bedarf eine der folgenden Tasten:

Taste	Bedeutung
CURSOR nach LINKS	Tastenbelegung definieren
CURSOR nach RECHTS	Funktionstasten mit String belegen
ALT	Tasten–Modus umschalten

7.7.2 Tastenbelegung definieren

Der beim Tastendruck erzeugte Zeichencode kann vom Benutzer umdefiniert werden. Jeder Taste sind, je nach dem, welche Umschalttaste gleichzeitig gedrückt wird, vier mögliche Definitionen zugeordnet: normal, Alpha–Shift, Shift und Control.

Normal–Modus: Taste wird nur allein gedrückt.

Alpha–Shift–Modus: Die Zuordnung der Tastenbelegung in diesem Modus wird mit der Commodore–Taste C= ein– und ausgeschaltet. Danach erscheint ein inverses Feld in der untersten Bildschirmzeile. Die erste Taste, die jetzt gedrückt wird, ist die zu definierende Taste. Die bisherige Zuordnung wird als eine Hex–Zahl in dem inversen Feld angezeigt. Durch Eingabe einer anderen Hex–Zahl kann der Taste die dieser Zahl entsprechende Zuordnung zugewiesen werden. Nachfolgend eine Übersicht der möglichen Tastenzuordnungen:

Code	Bedeutung
OOH	Null (keine Auswirkung)
01H bis 7FH	Normaler ASCII–Code
80H bis 9FH	Zugewiesene Strings
A0H bis AFH	80–Zeichen–Bildschirm: Zeichenfarbe
B0H bis BFH	80–Zeichen–Bildschirm: Hintergrundfarbe
C0H bis CFH	40–Zeichen–Bildschirm: Zeichenfarbe
D0H bis DFH	40–Zeichen–Bildschirm: Hintergrundfarbe
E0H bis EFH	40–Zeichen–Bildschirm: Rahmenfarbe
F0H	Disk–Status–Anzeige ein– oder ausschalten
F1H	Pause ein– oder ausschalten
F2H	Nicht definiert
F3H	
F4H	40–Zeichen–Bildschirm: horizontal rollen
F5H bis FFH	Nicht definiert

7.7.3 Tastenbelegung mit Strings definieren

Diese Funktion ermöglicht es, einer einzigen Taste nicht nur ein Zeichen, sondern auch eine Zeichenfolge (String) zuzuordnen. Die eingegebene Zeichenfolge wird in einem langen inversen Feld in der untersten Bildschirmzeile angezeigt.

Jede gedrückte Taste wird mit ihrer Bedeutung in den String übernommen. Damit können auch CTRL– und andere Steuerungszeichen in den String mit eingegeben werden. Um den Cursor innerhalb der eingegebenen Zeichenfolge zu bewegen und in der Zeile zu editieren, kann wie folgt verfahren werden: Zuerst sind die CTRL– und die rechte SHIFT–Taste gedrückt zu halten und dann gleichzeitig eine der folgenden Tasten zu betätigen:

Taste	Bedeutung
RETURN	String–Definition beenden
+ (obere Reihe)	Leerstelle einfügen
– (obere Reihe)	Zeichen unter Cursor löschen
CRSR links	Cursor nach links
CRSR rechts	Cursor nach rechts

7.7.4 ALT–Modus

Der ALT–Modus kann entweder ein– oder ausgeschaltet werden. Nach dem CP/M–Start ist er ausgeschaltet. Dieser Modus ermöglicht die Aussendung von 8–Bit–Zeichen.

7.7.5 Der CP/M–Bildschirm

Der CP/M–Bildschirm emuliert ein **ADM31**–Terminal. In der Tabelle 7.5 sind die Bildschirmsteuerfunktionen, die auch den Funktionen des älteren ADM 3A entsprechen, aufgeführt. Diese Funktionen sind eine Untermenge der ADM31–Funktionen. Die weiteren ADM31–Funktionen sind in den Tabellen 7.6 und 7.7 aufgeführt.

Tabelle 7.5 CP/M 3.0: Bildschirmsteuerfunktionen (ADM 3A entsprechend)

Funktion	Bedeutung
CTRL–G	Klingelzeichen
CTRL–H	Cursor nach links
CTRL–J	Cursor nach unten
CTRL–K	Cursor nach oben
CTRL–L	Cursor nach rechts
CTRL–M	Cursor an den Zeilenanfang setzen
CTRL–Z	Cursor Home und Bildschirm löschen
ESC = RC	Cursor an die Position RC setzen. R bestimmt die Zeile (Leerzeichen bis Ziffer 8) und C die Spalte (Leerzeile bis Buchstabe O) in Bezug auf die linke obere Ecke des Bildschirms.

Tabelle 7.6 CP/M 3.0: Bildschirmsteuerfunktionen (ADM31 entsprechend)

Funktion	Bedeutung
ESC T oder t	Löschen bis Zeilenende
ESC Y oder y	Löschen bis Bildschirmende
ESC : oder *	Cursor Home und Bildschirm löschen (auch die Statuszeile)
ESC Q	Zeichen einfügen
ESC W	Zeichen löschen
ESC E	Zeile einfügen
ESC R	Zeile löschen
ESC ESC ESC Farbe #	setzt die Bildschirmfarbe fest. Die Farbnummern liegen zwischen 0 und 15 und haben folgende Bedeutungen:
20H bis 2FH	Zeichenfarbe
30H bis 3FH	Hintergrundfarbe
40H bis 4FH	Rahmenfarbe (nur bei 40 Zeichen)

Tabelle 7.7 CP/M 3.0: Bildschirmsteuerfunktionen für einen 80–Zeichen–Bildschirm (ADM31 entsprechend)

Funktion	Bedeutung
EESC >	Halbe Intensität
ESC <	Volle Intensität
ESC G4	Inverse Darstellung ein
ESC G3	Unterstreichen einschalten (keine ADM31 Funktion)
ESC G2	Blinken einschalten
ESC G1	Alternativen Zeichensatz anwählen (Keine ADM31–Funktion)
ESC G0	Alle ESC–G–Attribute ausschalten

7.8 CP/M 3.0 – Befehlsübersicht

7.8.1 Struktur der CP/M–Befehle

Format: A > Befehlsschlüsselwort < Argument > < CR >

Zweck: Eine CP/M–Befehlszeile wird aus einem Befehlsschlüsselwort und einem frei wählbaren Befehlsargument zusammengesetzt und durch Drücken der RETURN–Taste (carriage return < CR >, deutsch: Wagenrücklauf) abgeschlossen und damit zur Ausführung an das CP/M–Betriebssystem abgesendet. Ein Befehlsschlüsselwort ist entweder ein residenter CP/M–Befehl (s. Tab. 7.8) oder der Name einer Programmdatei, die geladen und ausgeführt werden soll. Das frei wählbare Befehlsargument kann aus einer Laufwerksangabe, einer oder mehreren Dateiangaben und einigen Optionen oder Parametern bestehen.

7.8.2 CP/M–Befehlsarten

CP/M unterscheidet zwei Arten von Befehlen, die sog. residenten und die transienten Befehle:

Residente (built–in) Befehle: Interne Befehle, die im CP/M–Betriebssystem implementiert und immer abrufbar sind.

Transiente (externe) Befehle: Programme, die von der Diskette nachgeladen werden.

CP/M kennt sechs interne und über zwanzig transiente Befehle, die zum Standardlieferumfang von Digital Research gehören. Weitere Dienstprogramme, die unter CP/M lauffähig sind, können mitverwendet werden. Sie können auch mit Hilfe der bereitstehenden CP/M–Dienstprogramme eigene Programme erstellen.

Achtung:
Die Befehle LINK, MAC, RMAC, HEXCOM, SID, XREF, die in diesem Handbuch beschrieben sind, befinden sich nicht auf der mitgelieferten Utility-Diskette.

7.8.3 Residente Befehle

Tabelle 7.8 CP/M 3.0: Residente Befehle

Befehl	Bedeutung
DIR	Zeigt alle Dateinamen im Disketteninhaltsverzeichnis mit Ausnahme der mit dem SYS–Attribut markierten Dateien an.
DIRSYS	Zeigt die Namen der Dateien, die mit SYS–Attribut gekennzeichnet sind, an.
ERASE	Löscht den angegebenen Dateinamen aus dem Disketteninhaltsverzeichnis und kennzeichnet den belegten Platz wieder als frei.
RENAME	Ändert den Dateinamen.
TYPE	Zeigt den Inhalt von Textdateien auf dem Bildschirm an.
USER	Wechselt zwischen verschiedenen Benutzerbereichen.

7.8.4 Transiente (externe) Befehle

Diese Befehle sind in der nachfolgenden Tabelle 7.9 aufgelistet. Beim Aufruf dieser Befehle lädt CP/M die entsprechende Programmdatei von der Diskette in den Speicher und stellt alle weiteren Eingaben von der Befehlszeile dem aufgerufenen Programm in einem Puffer zur Verfügung.

Auch einige der internen Befehle existieren in einer erweiterten Form nur als transiente Befehle.

Wir finden hier die Befehle **DIR, ERASE, RENAME und TYPE.**

Wenn versucht wird, diese Befehle ohne Eingabe einer Dateispezifikation oder mit einem erweiterten Argument einzugeben, ohne die entsprechende Diskette im aktuellen Laufwerk zu haben, meldet das System

```
BEFEHL .COM required
```

als Zeichen, daß zum Ausführen diesen Befehls die entsprechende Diskettendatei erforderlich ist und auf der Diskette nicht gefunden werden konnte.

Das Wort **BEFEHL** steht hier für das entsprechende Befehlswort.

Beispiel:

```
A>RENAME
RENAME COM required
```

erscheint, wenn sich im Laufwerk A keine Diskette mit der Datei RENAME.COM befindet.

Tabelle 7.9 CP/M 3.0: Transiente Dienstprogramme

Befehl	Bedeutung
COPYSYS	Erzeugt eine neue bootfähige Diskette.
DATE	Setzt oder zeigt Datum und Zeit an.
DEVICE	Ordnet logische CP/M–Einheiten einer oder mehreren physikalischen Einheiten zu, ändert Baudrate und Übertragungsprotokoll oder setzt die Bildschirmgröße neu.

Tabelle 7.9 CP/M 3.0: Transiente Dienstprogramme (Fortsetzung)

Befehl	Bedeutung
DIR [o]	Zeigt das Disketteninhaltsverzeichnis und die Dateiattribute an.
DUMP	Zeigt den Inhalt einer Datei im ASCII– und Hexadezimalformat an.
ED	Erstellt und modifiziert Textdateien.
ERASE	Wird zum Löschen von Dateien mit den Platzhaltern * und ? verwendet.
FORMAT	Formatiert Disketten und erzeugt einen Boot–Sektor.
GET	Die Eingabe erfolgt vorübergehend aus einer Diskettendatei anstelle von der Tastatur.
HELP	Erklärt die CP/M–Befehle und deren Anwendungen.
HEXCOM	Erzeugt aus einer MAC–Ausgabedatei ein lauffähiges Programm.
INITDIR	Initialisiert das Disketteninhaltsverzeichnis für die Aufnahme von Zeit– und Datumsmarkierungen.
LIB	Bearbeitet Bibliotheksdateien der Typbezeichnung LIB.
LINK	Bindet REL–Programmodule zu lauffähigen Programmen zusammen.
MAC	Übersetzt Assembler–Quelldateien in Maschinencode.
PATCH	Modifiziert CP/M–Systemdateien.
PIP	Kopiert und kombiniert Dateien.
PUT	Leitet vorübergehend die Konsolen– oder Druckerausgabe in eine Diskettendatei um.

Tabelle 7.9 CP/M 3.0: Transiente Dienstprogramme (Fortsetzung)

Befehl	Bedeutung
RENAME	Ändert den Namen einer Diskettendatei, in Verbindung mit den Platzhalterzeichen auch mehrerer Dateien gleichzeitig.
RMAC	Übersetzt Assembler–Quelldateien in verschiebbare Programmodule.
SAVE	Kopiert Speicherinhalte in eine Diskettendatei.
SET	Setzt Dateioptionen, Disketten–Labels, Dateiattribute, Datums– und Zeitmarkierungen und den Paßwort-Schutz.
SETDEF	Setzt Systemoptionen, auch Laufwerkssuchverkettung.
SHOW	Zeigt Disketten und Laufwerkseigenschaften an.
SID	Testet Programme und ermöglicht das Auffinden von Programmablauffehlern.
SUBMIT	Führt automatisch mehrere Befehle aus.
TYPE	Gibt den Inhalt einer oder mehrerer (bei der Verwendung von * und ?) Textdateien auf dem Bildschirm (und Drucker, wenn angewählt) aus.
XREF	Erstellt Referenztabellen der Variablen von Assemblerprogrammen.

7.9 CP/M 3.0 – Befehlsbeschreibungen

In den Befehlsbeschreibungen in dieser Bedienungsanleitung wird folgende Syntax, wie in der Tabelle 7.10 aufgeführt, verwendet:

Tabelle 7.10 CP/M 3.0: Beschreibungssyntax

Zeichen	Bedeutung
{ }	Geschweifte Klammern schließen frei wählbare Befehlsteile ein.
|	Senkrechter Strich trennt alternative Befehlsteile in einer Befehlszeile voneinander.
<CR>	steht für die RETURN–Taste (carriage return).
↑	steht für die CONTROL–Taste.
n	ersetzt eine Zahl.
s	ersetzt eine Zeichenkette.
o	ersetzt eine Option oder Liste von Optionen.
[ ]	Eckige Klammern schließen eine Optionsliste ein.
( )	Runde Klammern schließen einen Bereich von Optionen innerhalb einer Optionsliste ein.
RW	Schreib/Lese–Attribut, d. h. kein Schreibschutz (Read–Write), das Gegenteil von RO.
RO	Schreibschutzattribut (Read–Only), das Gegenteil von RW.
SYS	Systemattribut, das Gegenteil von DIR.
DIR	Directoryattribut (Disketteninhaltsverzeichnis), das Gegenteil von SYS.

Tabelle 7.10 CP/M 3.0: Beschreibungssyntax (Fortsetzung)

Zeichen	Bedeutung
. . .	Das vorstehende Element kann so oft wiederholt werden, wie es gewünscht wird.
*	Jokerzeichen: Dieser Platzhalter ersetzt den ganzen oder einen Teil eines Dateinamen bzw. einer Typbezeichnung.
?	Jokerzeichen: Dieser Platzhalter ersetzt ein beliebiges einzelnes Zeichen an der Stelle in einem Dateinamen bzw. in einer Typbezeichnung, an der es steht.

COPYSYS–Befehl

Format: COPYSYS

Zweck: Der COPYSYS–Befehl kopiert das Betriebssystem CP/M 3.0 von einer CP/M–Systemdiskette auf eine andere Diskette. Die neue Diskette muß im selben Format wie die Originaldiskette formatiert sein.

Beispiel:

```
A>COPYSYS
```

Dieser Befehl hat keine Wirkung, bei Aufruf wird lediglich eine Meldung ausgegeben, daß beim C128 eine Diskette mit diesem Befehl nicht kopiert werden kann.

Verfahrensweise:

Neue Diskette formatieren (mit dem Programm FORMAT).

Kopieren mit 2 Laufwerken:

PIP B:=A:*.*AVOÜ (Ä und Ü bei deutschem Zeichensatz anstelle der rechteckigen Klammern).

Bei Verwendung eines Laufwerkes wird das virtuelle Laufwerk E benutzt.

DATE–Befehl

Format: DATE {CONTINUOUS}
DATE {Zeitangabe}
DATE SET

Zweck: Mit dem DATE–Befehl können Datum und Tageszeit angezeigt und gesetzt werden.

Beispiele:

```
A>DATE
```

zeigt das aktuelle Datum und die Uhrzeit an.

```
A>DATE C
```

zeigt Datum und Uhrzeit fortlaufend an.

```
A>DATE 05/26/85 11:30:00
```

setzt Datum und Uhrzeit.

```
A>DATE SET
```

setzt Datum und Uhrzeit im Dialog–Modus.

DEVICE–Befehl

Format:

Format:

```
DEVICE {NAMES | VALUES | pd | ld}
DEVICE ld = pd {o} {,pd {o},. . .}
DEVICE ld = NULL
DEVICE pd {o}
DEVICE CONSOLE [PAGE | COLUMNS = Spalten
                | LINES = Zeilen]
```

pd bedeutet physikalisches Gerät (physical device), darunter ist der aktuelle Gerätename einer vom System selektierten Geräteeinheit zu verstehen.

ld bedeutet logisches Gerät (logical device), darunter ist ein symbolischer Gerätename für eine Gruppe von Einheiten, die alle vom System bedient werden können, zu verstehen.

Zweck:

Der DEVICE–Befehl zeigt die aktuelle Zuweisung der logischen Geräte und der physikalischen Gerätenamen an. Außerdem können mit DEVICE die logischen Geräte den Peripheriegeräten des Rechners zugeordnet werden.

DEVICE legt auch das Übertragungsprotokoll und die Geschwindigkeit fest und zeigt die aktuelle Bildschirmgröße an oder legt sie fest.

Optionen:

Option	Funktion
XON	legt ein XON/XOFF–Übertragungsprotokoll fest.
NOXON	bedeutet, daß kein XON/XOFF–Protokoll verwendet wird. Der Rechner sendet unabhängig davon, ob das Empfangsgerät bereit ist oder nicht, Daten an das Gerät.
Baudrate	Die Baudrate legt die Übertragungsgeschwindigkeit des Gerätes fest. Das CP/M kann folgende Baudraten verwenden: 50 75 110 134 150 300 600 1200 1800 2400 3600 4800 7200 9600 19200
Beispiele:	A>DEVICE<CR>

```
Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xoff
KEYS    NONE  I     80COL  NONE  O     40COL  NONE  O
PRT1    NONE  O     PRT2   NONE  O     6551   19200 IOSX

Current Assignments:
CONIN:   = KEYS
CONOUT:  = 80COL
AUXIN:   = Null Device
AUXOUT:  = Null Device
LST:     = PRT1

Enter new assigment or hit RETURN
```

Diese Ausgabe zeigt die existierenden physikalischen Geräte und die aktuellen Zuweisungen der logischen Geräte des Systems an. Nach der Anforderung können Sie entweder eine neue Zuweisung eingeben oder mit RETURN die angezeigten bisherigen Zuordnungen übernehmen.

```
A>DEVICE NAMES<CR>

Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xoff
KEYS    NONE  I      80COL   NONE  O  40COL      NONE  O
PRT1    NONE  O      PRT2    NONE  O  6551       19200 IOSX
```

listet die physikalischen Geräte mit einer Kurzfassung der Geräteeigenschaften auf.

```
A>DEVICE VALUES<CR>

Current Assignments:
CONIN:    = KEYS
CONOUT:   = 80COL
AUXIN:    = Null Device
AUXOUT:   = Null Device
LST:      = PRT1
```

zeigt die aktuellen Zuweisungen der logischen Geräte an.

```
A>DEVICE CON<CR>
CONIN:    = KEYS
CONOUT:   = 80COL
```

zeigt die Zuordnungen der Konsole an.

```
A>DEVICE CONOUT:=80COL,PRT1<CR>
```

weist den Systemkonsolenausgang CONOUT: dem 80–Zeichen–Bildschirm und dem Drucker PRT1 zu.

```
A>DEVICE AUXIN:=6551[XON,300]<CR>
```

weist dem logischen Hilfseingang AUXIN: das physikalische Gerät 6551 (serielle Schnittstelle) zu, das ein XON/XOFF–Protokoll benutzt, und setzt die Übertragungsrate der Schnittstelle auf 300 Baud.

```
A>DEVICE LST:=NULL<CR>
```

unterbricht die LIST–Ausgabe des logischen Geräts LST:.

```
A>DEVICE 6551[NOXON,300]<CR>
```

schaltet das XON/–XOFF–Protokoll für das physikalische Gerät 6551 (serielle Schnittstelle) aus und setzt die Übertragungsrate auf 300 Baud.

```
A>DEVICE CON:[PAGE]<CR>

Console width set to 25 columns
Console page set to 24 lines
```

zeigt für die Konsole die aktuelle Seitenbreite in Spalten und die Seitenlänge in Zeilen an.

```
A>DEVICE CON:[COLUMNS=40,LINES=16]<CR>
```

setzt die Bildschirmgröße auf 40 Spalten und 16 Zeilen.

DIR–Befehl

Der DIR–Befehl zeigt das Disketteninhaltsverzeichnis (directory) an, d. h. die Dateinamen der bezeichneten Diskette und ggf. deren Eigenschaften.

Der DIR–Befehl hat drei verschiedene Ausführungsformen.

DIR
DIRS
DIR mit Optionen

DIR und DIRS sind residente, d. h. ins Betriebssystem integrierte (built in), Dienstprogramme. DIR mit Optionen ist ein transientes, d. h. externes, Dienstprogramm, das erst von einer Diskette in den Speicher geladen werden muß.

Residenter DIR–Befehl

Format:

DIR {d:}
DIR {dateispez}

DIRS {d:}
DIRS {dateispez}

Zweck:

Die residenten Befehle DIR und DIRS zeigen die Dateinamen an, die im Disketteninhaltsverzeichnis eingetragen sind. DIR gibt nur die Dateinamen des aktuellen Benutzerbereichs aus, die das DIR–Attribut besitzen, d. h. Dateien mit dem Systemattribut SYS werden nicht angezeigt. DIR akzeptiert die Jokerzeichen * und ? in einem Dateinamen.

Beispiele:

```
A>DIR<CR>
```

zeigt alle Dateien des Benutzerbereichs 0 im Laufwerk A an, die das DIR–Attribut besitzen.

```
A>DIR B:<CR>
```

zeigt alle Dateien des Benutzerbereichs 0 im Laufwerk B an.

```
2A>DIR C:ZIPPY.DAT<CR>
```

zeigt den Dateinamen ZIPPY.DAT an, falls sich diese Datei im Benutzerbereich 2 auf der Diskette im Laufwerk C befindet.

```
4A>DIR *.BAS<CR>
```

zeigt alle DIR–Dateien des Dateityps BAS im Benutzerbereich 4 von Laufwerk A an.

```
3B>DIR X*.C?D<CR>
```

zeigt alle Dateien des Benutzerbereichs 3 auf der Diskette im Laufwerk B an, deren Dateinamen mit dem Buch-

staben X beginnen und deren 3stelliger Dateitypname als erstes Zeichen C und als letztes Zeichen D enthält.

```
A>DIRS<CR>
```

zeigt alle Dateien mit dem Systemattribut SYS im Benutzerbereich 0 auf der Diskette im Laufwerk A an.

```
A>DIRS *.COM<CR>
```

zeigt alle SYS–Dateien des Typs COM auf der Diskette im Laufwerk A im Benutzerbereich 0 an. Auf eine Befehlsdatei COM des Benutzerbereichs 0 mit dem Systemattribut SYS kann im Suchketten–Modus (siehe SETDEF) aus jedem Benutzerbereich zugegriffen werden.

Externer DIR–Befehl mit Optionen

Format: DIR {d:} [o]
DIR {dateispez} {dateispez} . . .[o]

Zweck: Der externe DIR–Befehl mit Optionen ist eine erweiterte Version des residenten DIR–Befehls und zeigt die Dateien in besonderer Form an. DIR kann eine Datei von einem oder allen Laufwerken und aus einem oder allen Benutzerbereichen suchen. Die möglichen DIR–Optionen sind in Tabelle 7.11 aufgeführt.

Tabelle 7.11 CP/M 3.0: DIR–Optionen

Option	Funktion
ATT	zeigt die Dateiattribute an.
DATE	zeigt die Datums– und Zeiteinträge der Dateien an.
DIR	zeigt nur die Dateien mit DIR–Attribut an.
DRIVE = ALL	zeigt die Dateien aller angeschlossenen Laufwerke an.
DRIVE = (A,B,C,. . .,P)	zeigt die Dateien der bezeichneten Laufwerke an.
DRIVE = d	zeigt die Dateien des Laufwerks d an.
EXCLUDE	zeigt die Dateien an, die nicht in der Befehlszeile angegeben wurden.
FF	sendet ein Seitenvorschubszeichen (form feed) an den Drucker, wenn der Drucker mit CTRL–P aktiviert wurde.

Tabelle 7.11 CP/M 3.0: DIR–Optionen (Fortsetzung)

Option	Funktion
FULL	zeigt den Namen, die Größe, die Anzahl der 128–Byte-Records und die Attribute der Dateien an. Wenn eine Directory–Marke vorhanden ist, zeigt DIR den Paßwort-Schutzmodus und die Zeiteinträge an. Wenn keine Directory–Marke vorhanden ist, zeigt DIR zwei Dateieinträge pro Zeile an, indem die Spalten für das Paßwort und die Zeiteinträge ausgelassen werden. Die Anzeige wird alphabetisch sortiert. In der SET–Anweisung findet man eine Beschreibung über die Dateiattribute, die Directory–Marken, die Paßworte und den Schutz–Modus.
LENGTH = n	vereinbart eine Druckerausgabe von n Zeilen pro Seite. Die Kopfzeile der Directory wird auf jeder Seite wiederholt und in n nicht mitgezählt. n ist eine Zahl zwischen 5 und 65535.
MESSAGE	zeigt die Laufwerksnamen und die Benutzerbereiche an, die DIR gerade durchsucht.
NOPAGE	schaltet die seitenweise Ausgabe aus.
NOSORT	zeigt die Dateien in der Reihenfolge an, in der sie auf der Diskette stehen.
RO	zeigt nur die Dateien mit Schreibschutzattribut (Read–Only) an.
RW	zeigt nur die Dateien ohne Schreibschutzattribut (Read–Write) an.
SIZE	zeigt die Namen und die Größe der Dateien in KBytes (1024 Bytes) an.
SYS	zeigt nur die Dateien mit SYS–Attribut an.
USER = ALL	zeigt alle Dateien aller Benutzerbereiche im aktuellen oder bezeichneten Laufwerk an.

Tabelle 7.11 CP/M 3.0: DIR–Optionen (Fortsetzung)

Option	Funktion
USER = n	zeigt nur die Dateien des Benutzerbereichs n an.
USER = (0,1,2,. . .,15)	zeigt die Dateien der bezeichneten Benutzerbereiche an. Es können auch mehrere Bereiche angegeben werden.

Beispiele:

```
A>DIR C:[FULL]<CR>
```

zeigt alle Eigenschaften aller Dateien des Benutzerbereichs 0 im Laufwerk C an.

```
A>DIR C:[DATE]<CR>
```

gibt die Dateien im Laufwerk C mit ihren Zeiteinträgen aus.

```
A>DIR D:[RW,SYS]<CR>
```

zeigt alle Dateien des Benutzerbereichs 0 im Laufwerk D ohne Schreibschutz, aber mit dem System–Attribut SYS an.

```
3A>DIR[USER=ALL,DRIVE=ALL]<CR>
```

zeigt alle Dateien aller Benutzerbereiche (0 – 15) von allen Laufwerken an.

```
6B>DIR[EXCLUDE]*.DAT<CR>
```

gibt alle Dateien des Benutzerbereichs 6 von Laufwerk B aus, die nicht vom Dateityp DAT sind.

```
3B>DIR[SIZE]*.PLI *.COM *.ASM<CR>
```

zeigt alle Dateien des Benutzerbereichs 3 von Laufwerk B mit den Dateitypen PLI, COM und ASM an.

```
A>DIR[DRIVE=ALL,USER=ALL]TESTFILE.BOB<CR>
```

zeigt dann den Dateinamen TESTFILE.BOB an, wenn er in einem Laufwerk und in einem Benutzerbereich zu finden ist.

```
A>DIR[SIZE,RW]D:<CR>
```

gibt alle Dateien ohne Schreibschutz im Laufwerk D mit ihren Dateigrößen aus. Hierbei hat D: dieselbe Wirkung wie D:*.*.

DUMP–Befehl

Format: DUMP dateispez

Zweck: Der DUMP–Befehl zeigt den Inhalt einer Datei im Hexadezimal– und ASCII–Format an.

Beispiel:

```
A>DUMP DUMP.COM
CP/M 3 DUMP - Version 3.0
0000: C3 DB 03 00 00 00 00 00 ........
0008: 00 00 00 00 00 00 00 00 ........
0010: 44 55 4D 50 20 56 45 52 DUMP VER
0018: 53 49 4F 4E 20 33 2E 30 SION 3.0
```

ED–Befehl

Format: ED quell–dateispez {ziel–dateispez}

Zweck: ED ist ein zeilenorientierter Texteditor. Alle Editierfunktionen werden in einem Puffer im RAM durchgeführt. Der Text muß aus einer Quelldatei in den Puffer geladen und nach dem Beenden der Bearbeitung in die Quelldatei oder in eine Zieldatei zurückgeschrieben werden. ED löscht die Originalquelldatei nicht, sondern beläßt sie als sogenannte Sicherungs– bzw. Backup–Datei mit dem Dateityp BAK auf der Diskette. Um eine Datei umzuadressieren bzw. umzubenennen, müssen das Ziellaufwerk d und die Zieldatei angegeben werden. In Tabelle 7.12 sind die ED–Befehle aufgeführt.

Tabelle 7.12 CP/M 3.0: ED–Befehle

Befehl	Funktion
nA	Lädt n Zeilen aus der Quelldatei in den ED–Puffer.
0A	Lädt soviele Zeilen, bis die Hälfte des Puffers gefüllt ist.
#A	Lädt soviele Zeilen, bis der Puffer gefüllt oder das Dateiende erreicht ist.
B, –B	Positioniert den Cursor CP (Zeichenzeiger – character pointer) an den Anfang (B) oder das Ende (–B) des Puffers.
nC, –nC	Positioniert den Cursor CP um n Zeichen vorwärts (C) oder n Zeichen rückwärts (–C).
nD, –nD	Löscht n Zeichen vor (–D) oder hinter (D) der Cursorposition CP.
E	Beendet die Editierung, speichert die neue Datei auf der Diskette ab und springt zurück in CP/M.

Tabelle 7.12 CP/M 3.0: ED–Befehle (Fortsetzung)

Befehl	Funktion
Fs{↑Z}	Sucht die angegebene Zeichenkette (String) s im gesamtem Textpuffer.
H	Beendet die Editierung, speichert die neue Datei, ruft die neue Datei als neue Quelldatei auf und setzt mit dieser die Editierung fort.
I<CR>	Aufrufen des Einfügemodus für eine neue Textzeile nach der Cursorposition CP.
Is{↑Z}	Fügt die angegebene Zeichenkette (string) s an der aktuellen Cursorposition beginnend ein und positioniert den Cursor CP hinter das letzte Zeichen der Zeichenkette s.
Js1↑Zs2↑Zs3{↑Z}	Nebeneinanderstellen von Zeichenketten, indem die erste Zeichenkette gesucht wird, die zweite Zeichekette an die erste angefügt wird und alle Zeichen bis zur dritten Zeichenkette gelöscht werden.
nK, –nK	Löscht die folgenden (nK) bzw. die vorstehenden (–nK) n Zeilen von der Cursorposition aus.
nL, –nL, 0L	Positioniert den Cursor um n Zeilen vorwärts (nL) oder rückwärts (–nL). Mit 0L wird der Cursor an den Beginn der aktuellen Zeile gesetzt.
nMcommand	Führt die Befehle 'command' n–mal aus.
n, –n	Positioniert den Cursor um n Zeilen vorwärts (n) oder rückwärts (–n) und zeigt diese Zeile an.
n:	Positioniert den Cursor an den Beginn der Zeile n.
:ncommand	Führt den Befehl 'command' bis zur Zeile n aus.

Tabelle 7.12 CP/M 3.0: ED–Befehle (Fortsetzung)

Befehl	Funktion
Ns{↑Z}	Sucht die angegebene Zeichenfolge (string) s im Textpuffer.
O	Rücksprung in die Originaldatei.
nP, –nP	Positioniert den Cursor 23 Zeilen (Bildschirmseitenlänge) weiter und listet diese Zeilen auf dem Bildschirm auf.
Q	Verlassen der neuen Datei ohne Speicherung und Rücksprung ins CP/M–Betriebssystem.
R{↑Z}	Einlesen der Datei X$$$$$$$.LIB in den Textpuffer.
Rdateispez{↑Z}	Einlesen der Datei 'dateispez' in den Text– puffer.
Ss1↑Zs2{↑Z}	Ersetzen der Zeichenkette s1 durch s2.
nT,–nT, 0T	Listet n Zeilen auf.
U, –U	Wandelt alle folgenden Kleinbuchstaben nach der Eingabe von U in Großbuchstaben um. Diese Umwandlung wird mit –U beendet.
V, –V	Schaltet die Anzeige von Zeilennummern im Textpuffer mit der Eingabe von V ein. Diese Anzeige wird mit –V ausgeschaltet.
0V	Zeigt die verfügbare und die gesamte Größe des Textpuffers in Bytes (dezimal) an.
nW	Schreibt n Zeilen in die entsprechende Übergangsdatei mit der Typenkennung $$$.
0W	Schreibt soviele Zeilen in die Übergangsdatei, bis der Puffer halb voll ist.

Tabelle 7.12 CP/M 3.0: ED–Befehle (Fortsetzung)

Befehl	Funktion
nX{↑Z}	Kopiert die folgenden n Zeilen in die Datei X$$$$$$$.LIB.
nXdateispez{↑Z}	Kopiert die folgenden n Zeilen in die Datei {dateispez}. Wenn der X–Befehl schon vorher verwendet wurde, werden diese Zeilen daran angefügt.
0X{↑Z}	Löscht die Datei X$$$$$$$.LIB .
0Xdateispez{↑Z}	Löscht die Datei 'dateispez'.
nZ	Unterbricht die laufende ED–Operation um n Sekunden.

- Anmerkung: Der Cursor CP (Zeichenzeiger, Cursorposition) zeigt auf das aktuelle Zeichen im Textpuffer. Mit {↑Z} trennt man verschiedene ED–Befehle in derselben Zeile voneinander.

Beispiele:

```
A>ED TEST.DAT
A>ED TEST.DAT B:
>ED TEST.DAT TEST2.DAT
A>ED TEST.DAT B:TEST2.DAT
```

ERASE–Befehl

Format: ERASE {dateispez} {[CONFIRM]}

Zweck: Der ERASE–Befehl löscht eine oder mehrere Dateien aus dem Disketteninhaltsverzeichnis (directory). Die Platzhalter * und ? können verwendet werden. Das Disketteninhaltsverzeichnis und der Speicherplatz werden auf diese Weise für eine spätere Benutzung durch andere Dateien freigegeben.

ERASE kann mit ERA abgekürzt werden.

Option: [CONFIRM]

Diese Option bewirkt, daß vor dem Löschen einer Datei das CP/M–System eine Bestätigung bzw. Quittierung verlangt. CONFIRM kann mit C abgekürzt werden.

Beispiele:

```
A>ERASE X.PAS<CR>
```

löscht die Datei X.PAS von der Diskette im Laufwerk A.

```
A>ERA *.PRN [CONFIRM]<CR>
Confirm (Y/N)?Y
```

Alle Dateien des Typs PRN werden von der Diskette im Laufwerk A gelöscht.

```
A>ERA A:MY*.* [CONFIRM]<CR>
```

Vor dem Löschen der Dateien auf der Diskette im Laufwerk A, deren Namen mit MY anfangen, wird mit einem Fragezeichen eine Quittierung angefordert.

GENCOM–Befehl

Format: GENCOM {COM–dateispez} {RSX–dateispez}. . .
{[LOADER | NULL | SCB = (o,n)]}

Zweck: Der GENCOM–Befehl erzeugt eine spezielle COM–Datei mit angehängten RSX–Dateien. Der GENCOM–Befehl kann auch aus einer mit GENCOM erzeugten wieder die ursprüngliche COM–Datei ohne den am Anfang stehenden 'Header' und den RSX–Dateien erzeugen. GENCOM kann auch Header–Records an eine COM–Datei anfügen.

Tabelle 7.13 CP/M 3.0: GENCOM–Optionen

Option	Funktion
LOADER	fügt ein 256–Byte Header–Record hinzu und setzt ein Loader–Flag.
NULL	zeigt an, daß nur RSX–Dateien angegeben werden. GENCOM erzeugt eine leere COM–Datei für die RSX–Dateien. Als Name der erzeugten COM–Datei wird der Dateiname der ersten RSX–Dateibezeichnung verwendet.
SCB = (o,n)	setzt den System–Control–Block des Programms aus den angegebenen Hexadezimalzahlen (o,n).

Beispiele:

```
A>GENCOM MYPROG PROG1 PROG2
```

erzeugt eine neue COM–Datei MYPROG.COM mit den angefügten RSX–Dateien PROG1 und PROG2.

```
A>GENCOM PROG1 PROG2[NULL]
```

erzeugt eine COM–Datei PROG1.COM mit den RSX–Dateien PROG1 und PROG2.

```
A>GENCOM MYPROG
```

GENCOM lädt MYPROG, entfernt den Header, löscht alle angefügten RSX–Dateien und lädt es wieder im originalen COM–Format zurück.

```
A>GENCOM MYPROG PROG1 PROG2
```

GENCOM untersucht die schon mit GENCOM erzeugte Datei MYPROG.COM danach, ob PROG1.RSX und PROG2.RSX schon als RSX–Dateien angefügt wurden. Wenn eine von beiden schon angefügt wurde, ersetzt GENCOM diese durch die neue RSX-Datei, anderenfalls fügt GENCOM die bezeichnete RSX-Datei an die COM-Datei an.

GET–Befehl

Format: GET {CONSOLE INPUT FROM} FILE dateispez
↑[{ECHO ↑ NO ECHO} ↑ SYSTEM]}

GET {CONSOLE INPUT FROM} CONSOLE

Zweck: Der GET–Befehl steuert das Betriebssystem in der Weise, daß die Konsoleneingabe für den nächsten Systembefehl oder für ein über die Konsole eingegebenes Benutzerprogramm von einer Datei erfolgen kann.

Die Konsoleneingabe wird von einer Datei genommen, bis das Programm endet. Wenn die Datei abgearbeitet ist, ehe die Programmeingabe beendet wurde, erwartet das Programm eine nachfolgende Eingabe über die Konsole. Wenn das Programm endet, ehe alle Eingaben abgearbeitet wurden, springt das System zur Konsoleneingabe zurück.

Mit der SYSTEM–Option geht das System für eine Konsoleneingabe sofort auf die bezeichnete Datei. Das System verlangt eine Konsoleneingabe, wenn das Dateiende erreicht wurde. Mit dem Befehl GET CONSOLE INPUT FROM CONSOLE als Befehlszeile in der Eingabedatei springt das System für eine Konsoleneingabe zurück zur Konsole.

Tabelle 7.14 CP/M 3.0: GET–Optionen

Option	Bedeutung
ECHO	bedeutet, daß eine Eingabe auf der Konsole angezeigt wird. Dies ist eine voreingestellte Option.
NO ECHO	bedeutet, daß die Dateieingabe nicht auf der Konsole angezeigt wird. Die Programmausgabe und die Systembereitschaftsmeldungen werden nicht von dieser Option berührt und deshalb auf der Konsole angezeigt.
	SYSTEM bedeutet, daß alle Systemeingaben sofort aus der in der Befehlszeile angegebenen Diskettendatei gelesen werden. GET nimmt die System– und Programmeingabe aus der Datei, bis die Datei abgearbeitet wurde oder bis ein GET–Konsolenbefehl aus der Datei gelesen wird.

Beispiele:

```
A>GET FILE XINPUT
A>MYPROG
```

übermittelt dem System, das GET–Dienstprogramm zu aktivieren. Da SYSTEM nicht angegeben wurde, liest das System die nächste Eingabezeile von der Konsole und führt MYPROG aus. Wenn das Programm MYPROG eine Konsoleneingabe fordert, wird diese von der Datei XINPUT genommen. Wenn MYPROG endet, springt das System für eine Konsoleneingabe zurück zur Konsole.

```
A>GET FILE ABCD [SYSTEM]
```

steuert das System so, daß eine nachfolgen– de Konsoleneingabe aus der Datei ABCD gelesen wird, weil die SYSTEM–Option im Befehl enthalten ist. Das System springt für eine Konsoleneingabe zurück zur Konsole, wenn das Ende der Datei ABCD erreicht wird. Alternativ kann ABCD das System zurück zur Konsole steuern, wenn ABCD einen Befehl GET CONSOLE erhält.

```
A>GET CONSOLE
```

teilt dem System mit, eine Konsoleneingabe von der Konsole zu holen. Dieser Befehl kann in einer Datei benutzt werden, die vorher in einem Befehl GET FILE benannt wurde, und die bereits vom System für eine Konsoleneingabe gelesen wird. Der Befehl wird benutzt, um die Konsoleneingabe zurück zur Konsole zu legen, ehe das Dateiende erreicht wird.

HELP–Befehl

FORMAT: HELP {topic}{subtopic1. . .subtopic8}[NOPAGE | LIST]}

Zweck: Das HELP–Programm zeigt eine Liste von CP/M–Befehlen an und liefert auf Abruf Informationen darüber.

HELP topic informiert über diesen Begriff.

HELP topic subtopic informiert über diesen Unterbegriff.

Ein oder zwei Buchstaben genügen zur Identifizierung der Begriffe. Wird nur HELP eingegeben, erscheint der Text HELP> mit einer Liste von Unterbegriffen auf dem Bildschirm. Die Eingabe eines

- Fragezeichens ? gibt die Liste der Hauptbegriffe aus.

- Punktes und Unterbegriffnamens gibt den Unterbegriff aus.

- Punktes wiederholt die letzte Ausgabe.

Das Drücken der RETURN–Taste bewirkt einen Rücksprung in CP/M–Betriebssystem.

Die Option [NOPAGE] schaltet die seitenweise (mit 24 Zeilen) Ausgabe aus.

Das Drücken einer beliebigen Taste beendet die Anzeige und zeigt wieder den Text HELP> an.

Beispiele:

```
A>HELP
A>HELP DATE
A>HELP DIR OPTIONS
HELP>.OPTIONS
HELP>SET
HELP>SET PASSWORD
HELP>.PASSWORD
HELP>.
HELP><CR>
```

Verfügbare Begriffe und Bezeichnungen:

COMMANDS	CNTRLCHARS	COPYSYS	DATE	DEVICE	DIR
DUMP	ED	ERASE	FILESPEC	GENCOM	GET
HELP	HEXCOM	INITDIR	LIB	LINK	MAC
PATCH	PIP (COPY)	PUT	RENAME	RMAC	SAVE
SET	SETDEF	SHOW	SID	SUBMIT	TYPE
USER	XREF				

HEXCOM–Befehl

Format: HEXCOM filename

Zweck: Der HEXCOM–Befehl erzeugt aus einer HEX–Datei eine direkt lauffähige Befehlsdatei vom Dateityp COM. Der Name des transformierten Programms ist derselbe wie der der eingegebenen Datei, die Typbezeichnung jedoch COM. HEXCOM verlangt immer eine Eingabe mit dem Dateityp HEX.

Beispiel:

```
A>HEXCOM B:PROGRAM
```

erzeugt eine Befehlsdatei PROGRAM.COM aus der eingegebenen HEX–Datei PROGRAM.HEX.

INITDIR–Befehl

Format: INITDIR {d:}

Zweck: Der INITDIR–Befehl bereitet ein Disketteninhaltsverzeichnis für Datums– und Zeiteinträge zu den Dateien auf der Diskette vor. Ist die Diskette bereits initialisiert, so kann man mit INITDIR sämtliche Zeit– und Datumseinträge löschen und den Bereich wieder freigeben.

Beispiel:

```
A>INITDIR
ERROR: Unrecognized drive.
Drive: x

Enter Drive: a

INITDIR WILL ACTIVATE TIME STAMPS FOR SPECIFIED
DRIVE
Do you want to re-format the directory on drive:
A (Y/N)?Y
```

INITDIR aktiviert den Zeit–Datums–Bereich auf der Diskette im bezeichneten Laufwerk.

LIB–Befehl

Format: LIB dateispez {[I : M : P : D]}

LIB dateispez {[I : M : P]} = dateispez {modifier} {,dateispez {modifier}. . .}

Zweck: Man benutzt das LIB–Dienstprogramm, um Bibliotheksdateien zu erzeugen und um Objektmodule bei einer vorhandenen Bibliotheksdatei anzufügen, zu ersetzen, auszulesen, zu löschen oder auch Kenntnis über den Inhalt einer Bibliotheksdatei zu erhalten.

Mit LIB erzeugt und pflegt man Bibliotheksdateien, die Objektmodule im Microsoft–REL– Dateiformat enthalten. Diese Module werden vom Digital–Research–Makro–Assembler RMAC oder von einigen Compilern erzeugt.

Man kann den LINK–Befehl verwenden, um Objektmodule aus einer Bibliotheksdatei mit anderen Objektmodulen zu verbinden. LINK wählt selbsttätig aus der Bibliotheksdatei nur jene Module aus, die von den zu linkenden Moduln benötigt werden, und bildet daraus eine ablauffähige Datei mit dem Dateityp COM.

Tabelle 7.15 CP/M 3.0: LIB–Optionen

Option	Bedeutung
I	Die INDEX–Option erzeugt eine indizierte Bibliotheksdatei vom Typ IRL. LINK sucht zuerst nach indizierten Bibliotheksdateien und dann erst nach nicht indizierten.
M	Die MODUL–Option zeigt die Modulnamen an.
P	Die PUBLICS–Option zeigt die Modulnamen und die global verwendeten Variablen der neuen Bibliotheksdatei an.
D	Die DUMP–Option zeigt den Inhalt des Objektmoduls im ASCII–Format an.

Modifier:

Man verwendet einen Modifier in der Befehlszeile, um mit LIB Module in einer Bibliotheksdatei zu löschen, zu ersetzen oder auszuwählen. Spitze Klammern <..> schließen die zu löschenden oder zu ersetzenden Module ein. Runde Klammern (..) schließen die ausgewählten Module ein.

Tabelle 7.16 CP/M 3.0: LIB–Modifier

Funktion	Modifier
Löschen	<module=>
Ersetzen	<module=dateiname.REL>
	Wenn Modulname und Dateiname gleich sind, kann folgende Kurzform verwendet werden:
	<dateiname>
Auswählen	(modERST–modLETZT,mod1,mod2,...,modN)

Beispiele:

```
A>LIB TEST4[P]
```

zeigt alle Module und globalen Variablen von TEST4.REL an.

```
A>LIB TEST5[P]=FILE1,FILE2
```

erzeugt TEST5.REL aus FILE1.REL und FILE2.REL und zeigt alle Module und globalen Variablen von TEST5.REL an.

```
A>LIB TEST=TEST1(MOD1,MOD4),TEST2(C1-C4,C6)
```

erzeugt eine Bibliotheksdatei TEST.REL aus den Modulen zweier Quelldateien. TEST1.REL liefert MOD1 und MOD4. LIB wählt weiterhin aus TEST2 die Module C1 und C4 sowie alle dazwischen liegenden Module und das Modul C6 aus.

```
A>LIB FILE2=FILE3<MODA=>
```

erzeugt FILE2.REL aus FILE3.REL, jedoch ohne MODA, das ein Modul in FILE3.REL ist.

```
A>LIB FILE6=FILE5<MODA=FILEB.REL>
```

erzeugt FILE6.REL aus FILE5.REL, wobei MODA durch FILEB.REL ersetzt wird.

```
A>LIB FILE6=FILE5<NAME>
```

Modul NAME befindet sich in FILE5.REL. Wenn LIB die Datei FILE6.REL aus FILE5.REL erzeugt, wird das Modul NAME durch die namensgleiche Datei NAME.REL ersetzt.

```
A>LIB FILE1[I]=B:FILE2
(PLOTS,FIND,SEARCH-DISPLAY)
```

erzeugt FILE1.IRL auf dem Laufwerk A aus den ausgewählten Modulen PLOTS, FIND und SEARCH bis DISPLAY, alle aus FILE2.REL auf dem Laufwerk B.

LINK–Befehl

Format: LINK d:{dateispez,{[o]}=}dateispez{[o]} {,. . .}

Zweck: LINK verbindet verschiebbare Objektmodule, die z. B. von RMAC und PL/I–80 erzeugt werden, zu einer direkt ausführbaren COM–Datei. Verschiebbare Dateien können externe Referenzmodule und globale Variable enthalten und Bezug auf Module in Bibliotheksdateien nehmen. LINK untersucht die Bibliotheksdateien und fügt die Referenzmodule in die ausgegebene Datei ein. Nähere Informationen über LINK sind im entsprechenden Handbuch (Programmer's Utilities Guide) zu finden.

Optionen: Eine LINK–Option schaltet die Parameter ein, die den Ablauf beeinflussen. LINK–Optionen folgen den Dateibezeichnungen 'dateispez' und sind in rechteckige Klammern eingeschlossen. Mehrere Optionen werden durch Kommata getrennt.

Tabelle 7.17 CP/M 3.0: LINK–Optionen

Option	Bedeutung
A	Zusätzlicher Speicher. Der Linker reduziert den Pufferbereich und schreibt Zwischendaten auf die Diskette.
B	Erstellen eines BIOS im gebankten CP/M– System: 1. Anfügen eines DATA–Abschnitts an die Page–Grenze. 2. Ablegen der Länge von Kodeteilen im Header'. 3. Vorgaben für den Dateityp SPR.
Dhhhh	DATA–Anfang. Die Anfangsadresse für den COMMON– und DATA–Speicherbereich wird festgelegt.

Tabelle 7.17 CP/M 3.0: LINK–Optionen (Fortsetzung)

Option	Bedeutung
Gn	Startadresse (go). Die Startadresse für das Programm wird auf das Label n gelegt.
Lhhhh	Laden. Die vorgegebene Ladeadresse eines Moduls kann statt auf 0100H (Voreinstellung) auf den Wert hhhh gelegt werden.
Mhhhh	Speichergröße. Ein zusätzlicher freier Speicherbereich kann für MP/M–Module vereinbart werden.
NL	Kein Listing. Die Symboltabelle wird nicht auf der Konsole ausgegeben.
NR	Kein Record. Es wird keine Datei mit Symboltabelle erzeugt.
OC	Ausgabe einer COM–Datei. Die Objektdatei ist vom Dateityp COM. Dies ist der Normalfall.
OP	Ausgabe einer PRL–Datei. Die verschiebbare Objektdatei (für einen Ablauf unter MP/M in verschiebbaren Modulen) ist vom Typ PRL.
OR	Ausgabe einer RSP–Datei für MP/M.
OS	Ausgabe einer SPR–Datei für MP/M.
Phhhh	Programmbasisadresse. Die Basisadresse für den Programmkode kann statt auf 0100H (Voreingestellung) auf den Wert hhhh gelegt werden.

Tabelle 7.17 CP/M 3.0: LINK–Optionen (Fortsetzung)

Option	Bedeutung
Q	Die Symbole mit einem Fragezeichen als erstes Zeichen werden mit in die Symboltabelle aufgenommen.
S	Suchen. Es werden nur die notwendigen Teile der vorstehenden Datei geladen. Die S–Option wird bei Bibliotheksdateien verwendet.
$Cd	Die Konsolenausgabe wird für d = X zur Konsole geleitet, für d = Y zum Drucker geleitet oder für d = Z unterdrückt. Die Voreinstellung ist X.
$Id	Die Zwischendateien werden in Laufwerk d (A – E) gesucht. Die Voreinstellung ist das aktuelle Laufwerk.
$Ld	Die Bibliotheksdateien werden im Laufwerk d (A – E) gesucht. Die Vorgabe ist das aktuelle Laufwerk.
$Od	Die Objektdatei wird auf das Laufwerk d (A – E) geschrieben. Für d = Z wird die Ausgabe unterdrückt. Die Voreinstellung ist das Laufwerk, in dem die erste Datei des LINK–Befehls steht.
$Sd	Die Symboldatei wird auf das Laufwerk d (A – E) geschrieben. Für d = Y geht die Ausgabe zum Drucker, und für d = Z wird sie unterdrückt.
Beispiele	`A>LINK B:MYFILE[NR]` LINK auf Laufwerk A verwendet als Eingabedatei MYFILE.REL auf Laufwerk B und erzeugt die ausführbare Programmdatei MYFILE.COM auf Laufwerk B. Die NR–Option bedeutet, daß keine Datei mit Symboltabelle erzeugt wird.

```
A>LINK M1,M2,M3
```

LINK verbindet die einzelnen kompilierten Dateien M1, M2 und M3, berechnet ihre externen Referenzmodule und erzeugt die ausführbare Programmdatei M1.COM.

```
A>LINK M=M1,M2,M3
```

LINK verbindet die einzelnen kompilierten Dateien M1, M2 und M3 und erzeugt die ausführbare Programmdatei M.COM.

```
A>LINK MYFILE,FILE5<S>
```

Die S–Option bewirkt, daß LINK die Datei FILE5 als eine Bibliotheksdatei sucht. LINK verbindet MYFILE.REL mit den Referenzunterprogrammen aus FILE5.REL auf dem aktuellen Laufwerk A und erzeugt MYFILE.COM auf Laufwerk A.

MAC–Befehl

Format: MAC FILENAME {$OPTIONEN}

Zweck: Der CP/M 3.0 Macro–Assembler MAC liest Anweisungen in Assemblersprache aus einer Eingabedatei vom Typ ASM, assembliert diese Anweisungen und erstellt drei Ausgabedateien mit dem Eingabedateinamen und den Typbezeichnungen HEX, PRN und SYM.

FILENAME.HEX enthält den Objectcode im INTEL–Hexadezimal–Format.

FILENAME.PRN enthält das kommentierte Quell–Listing, das entweder ausgedruckt oder auf der Konsole angezeigt werden kann.

FILENAME.SYM enthält die sortierte Tabelle der im Programm definierten Symbole.

Optionen: Mit den optionellen Angaben können die Eingabe– und Ausgabelaufwerke oder –einheiten bestimmt werden. Die möglichen Optionen sind in den Tabellen 7.18 und 7.19 aufgeführt.

Tabelle 7.18 CP/M 3.0: MAC–Input–Output–Optionen

Option	Bedeutung
A	Quell–Laufwerk für die ASM–Datei (A – E).
H	Ziellaufwerk für die HEX–Datei (A – E, Z).
L	Quell–Laufwerk für die LIB–Dateien, die mit dem MACLIB Statement aufgerufen werden (A – E).
P	Zieleinheit für die PRN–Datei (A – E, X, P, Z).
S	Zieleinheit für die SYM–Datei (A – O, X, P, Z).

A – E sind die möglichen Laufwerke (sofern vorhanden),
X leitet die entsprechende Ausgabe auf die Konsole,
P auf den Drucker und Z unterdrückt die Ausgabe.

Tabelle 7.19 CP/M 3.0: MAC–Ausgabemodifikationen

Modif.	Bedeutung
+L	Input–Zeilen, die von Makrobibliotheksdateien LIB gelesen werden, auflisten.
–L	Auflistung der Input–Zeilen, die von Makrobibliotheksdateien LIB gelesen werden, unterbinden (Voreinstellung).
+M	Alle Makrozeilen, die beim Assemblieren erzeugt werden, auflisten.
–M	Die Auflistung der Makrozeilen, die beim Assemblieren erzeugt werden, unterbinden.

Tabelle 7.19 CP/M 3.0 Modifikationen der Ausgabedateien (Fortsetzung)

Modif.	Bedeutung
*M	Makroerweiterungen nur im HEX–Format auflisten.
+Q	Alle LOCAL–Symbole in der Symboltabelle auflisten.
–Q	Die Auflistung aller LOCAL–Symbole in der Symboltabelle unterbinden.
+S	Symboldatei an die PRN–Datei anhängen.
–S	Die Erzeugung der Symboldatei unterbinden.
+1	Erzeugen eines Pass–1–Listings für 'macro debugging' in der PRN Datei.
–1	Pass–1–Listing unterdrücken (Voreinst.)

Beispiele:

```
A>MAC SAMPLE
```

Der Makroassembler MAC wird vom Laufwerk A aufgerufen und bearbeitet eine Datei SAMPLE.ASM, die auch auf der Diskette im Laufwerk A sein muß. Die Ausgabedateien SAMPLE.OBJ, SAMPLE.PRN und SAMPLE.SYM werden auch im Laufwerk A abgelegt.

```
A>MAC SAMPLE $PB AA HB SX
```

In diesem Beispiel folgt dem MAC–Aufruf und dem Quelldateinamen eine optionelle Parameterliste. Die PRN–Datei wird auf das Laufwerk B geleitet, die ASM-Datei vom Laufwerk A gelesen, die HEX–Datei wird auf Laufwerk B abgelegt und die Symboltabelle wird auf der Konsole gelistet. Als Trennzeichen zwischen einzelnen Parametern werden Leerstellen benutzt.

PATCH–Befehl

Format: PATCH FILENAME {Typ} {n}

Zweck: Der PATCH–Befehl zeigt auf dem Bildschirm die Nummer des bereits durchgeführten 'Patches' an oder führt ihn, falls noch nicht geschehen, aus.

PATCH kann nur auf CP/M 3.0–Systemdateien angewendet werden. Die zulässigen Typbezeichnungen sind COM, PRL oder SPR. Ohne Angabe der Typbezeichnung sucht CP/M nach einem COM–Dateinamen.

Die Patch–Nummer darf nur zwischen 1 und 32 liegen.

Beispiel:

```
A>PATCH SHOW 2
```

Die Datei SHOW.COM wird mit dem Patch Nr. 2 bearbeitet. Auf der Konsole wird folgende Meldung ausgegeben:

```
Do you want to indicate that Patch #2 has been
installed for SHOW.COM?Y
```

Wird der Patch erfolgreich durchgeführt, wird auf der Konsole angezeigt:

```
Patch Installed
```

Wird der Patch nicht erfolgreich durchgeführt, erscheint folgende Meldung auf der Konsole:

```
Patch not Installed
```

Eine der nachfolgend aufgeführten möglichen Fehlermeldungen wird ausgeben:

```
* ERROR: Patch requires CP/M 3.
* ERROR: Invalid filetype typ.
* ERROR: Serial Number mismatch.
* ERROR: Invalid patch number n.
```

PIP–Befehl

Format: ZIEL = QUELLE

PIP | d:{dateispez} {[Gn]} = dateispez{[o]} {,..} |

ZIEL–Angabe = d: {[o]}

PIP

Zweck: Das Dateikopierprogramm PIP kann Dateien zwischen Disketten, Druckern, Konsolen oder anderen logischen Einheiten kopieren, kombinieren oder übertragen. Wenn als Quelldateiname mehrere mit Komma getrennte Namen aufgeführt werden, werden diese Dateien in eine Zieldatei zusammenkopiert. {[o]} ist eine Kombination der verfügbaren Optionen. Die {[Gn]}–Option bestimmt, unter welcher Benutzerbereichsnummer die Zieldatei abgelegt wird. Wird nur der Befehl PIP ohne zusätzliche Spezifikationen angegeben, wird ein * als Prompt–Zeichen angezeigt und die Eingabe einer Befehlszeile erwartet.

Als ZIEL oder QUELLE kann jede sinnvolle CP/M 3.0 logische Einheit angegeben werden.

Wenn nur ein Einzellaufwerk zur Verfügung steht, wird als Ziel–Laufwerk das virtuelle Laufwerk E angegeben. Während des Kopierens erscheinen dann Aufforderungen zum Wechseln der Quell– und Zieldisketten.

Beispiele: Eine Datei von einer auf eine andere Diskette kopieren:

```
A>PIP b:=a:draft.txt
A>PIP b:draft.txt=a:
B3>PIP myfile.dat=A:[G9]
A9>PIP B:[G3]=myfile.dat
```

Eine Datei kopieren und umbenennen:

```
A5>PIP newdraft.txt=olddraft.txt
C8>PIP b:newdraft.txt=a:olddraft.txt
```

Mehrere Dateien auf einmal nacheinander einzeln kopieren:

```
A>PIP b:=draft.*
C>PIP b:=*.txt[G5]
C>PIP a:=*.com<wr>
B>PIP a:<g3>=c:*.*
```

Mehrere Dateien zusammenkopieren:

```
A>PIP b:new.dat=file1.dat,file2.dat
```

Kopieren, Umbenennen und Zuordnen im Benutzerbereich 1:

```
A>PIP newdraft.txt<g1>=olddraft.txt
```

Kopieren und Umbenennen aus dem Benutzerbereich 1:

```
A>PIP newdraft.txt=olddraft.txt[g1]
```

Kopieren von und zu logischen Einheiten:

```
A>PIP b:funfile.sue=con:
A>PIP lst:=con:
A>PIP lst:=b:draft.txt[t8]
A>PIP prn:=b:draft.txt
```

Tabelle 7.20 CP/M 3.0: PIP–Optionen

Option	Bedeutung
A	Archiv. Nur die Dateien kopieren, die seit dem letzten Kopieren verändert wurden.
C	Confirm. Wenn mehrere Dateien auf einmal kopiert werden, wird jedesmal eine Bestätigung verlangt.
Dn	Ausblenden aller Zeichen nach der n–ten Stelle in einer Zeile, z. B. Begrenzen der Zeilenlänge bei der Druckerausgabe auf die druckbare Zeilenlänge.

Tabelle 7.20 CP/M 3.0: PIP–Optionen (Fortsetzung)

Option	Bedeutung
E	Echo. Alle Zeichen werden während der Übertragung auch auf der Konsole angezeigt.
F	Filter (form feeds). Alle Seitenvorschubszeichen (ASCII–Wert 0CH oder CTRL–L) werden beim Kopieren entfernt.
Gn	Kopieren in oder aus dem Benutzerbereich n.
H	Überprüfen der Dateien auf korrekte INTEL– Hex–Formate bei der Übertragung von HEX–Dateien. Fehler werden auf der Konsole angezeigt.
I	:00–Records im INTEL–Hex–Format werden ignoriert. Mit der I–Option wird automatisch auch die H–Option aktiviert.
L	Großgeschriebene (A – Z) werden in kleingeschriebene Zeichen (a – z) umgewandelt. Alle anderen Zeichen bleiben unverändert.
N	Zeilennummer an den Anfang jeder Zeile setzen. Die Nummern beginnen mit 1 und werden von Zeile zu Zeile um 1 erhöht. Nach der Nummer folgt ein Doppelpunkt.
O	Kopieren von Objektdateien. Das Dateiendzeichen CTRL–Z (↑Z) wird wie ein normales Zeichen übertragen und beendet die Übertragung nicht.
Pn	Seitenlänge setzen. Nach n Zeilen wird ein ASCII–Wert 0CH (Seitenvorschub) eingefügt.
Qs↑Z	Beenden des Kopierens nach dem String s.
R	Kopieren von Systemdateien mit dem Attribut SYS.

Tabelle 7.20 CP/M 3.0: PIP–Optionen (Fortsetzung)

Option	Bedeutung
Ss↑Z	Die Datei wird erst ab String s kopiert.
Tn	TAB–Zeichen (ASCII–Wert 09H) werden durch n Leerzeichen ersetzt.
U	Kleingeschriebene (a – z) werden in großgeschriebene Zeichen (A – Z) umgewandelt. Alle anderen Zeichen bleiben unverändert.
V	Verify. Kopierte Daten werden auf Korrektheit überprüft (nur beim Kopieren auf Disketten möglich).
W	Eine existierende Nur–Lese–Datei mit dem RO–Attribut (Read Only) wird beim Kopieren auf der Zieldiskette überschrieben.

PUT–Befehl

Format

```
PUT CONSOLE {OUTPUT TO} dateispez {OPTION}
PUT PRINTER {OUTPUT TO} dateispez {OPTION}
PUT CONSOLE {OUTPUT TO} CONSOLE
PUT PRINTER {OUTPUT TO} PRINTER
```

Zweck:

Mit dem PUT–Befehl wird die Konsolen– oder Druckerausgabe in eine Diskettendatei umgeleitet. Diese Umleitung beginnt mit dem nächsten eingegebenen Befehl und wird mit einem entsprechenden PUT–Befehl beendet.

PUT mit der SYSTEM–Option leitet alle nachfolgenden Konsolen– oder Druckerausgaben in die angegebene Datei. Diese Option wird beendet mit der Eingabe PUT CONSOLE oder PUT PRINTER.

Optionen:

Folgende Optionen sind möglich:

```
[{ECHO / NO ECHO} {FILTER / NO FILTER? | {SYSTEM}]
```

Tabelle 7.21 CP/M 3.0: PUT–Optionen

Option	Bedeutung
ECHO	Die Ausgabe wird gleichzeitig auf der Konsole angezeigt (Voreinstellung).
NO ECHO	Die Ausgabe auf der Konsole wird unterdrückt.
FILTER	CTRL–Zeichen werden in druckbare Zeichen umgewandelt. Beispielsweise wird das ESC– Zeichen als ↑[wiedergegeben.
NO FILTER	CTRL–Zeichen werden nicht umgewandelt (Voreinstellung).
SYSTEM	Die SYSTEM– und PROGRAMM–Ausgaben werden in die angegebene Datei geschrieben, bis ein PUT–CONSOLE– oder PUT–PRINTER–Befehl die Ausgabe wieder zurückleitet.

Beispiele:

```
A>PUT CONSOLE OUTPUT TO FILE XOUT [ECHO]
```

Leitet Konsolenausgabe in eine Datei XOUT. Die Ausgabe wird auf der Konsole wiederholt.

```
A>PUT PRINTER OUTPUT TO FILE XOUT
A>TESTPROG
```

Leitet die Druckerausgabe des Programms TESTPROG in eine Datei XOUT. Die Ausgabe wird auf dem Drucker nicht wiederholt.

```
A>PUT PRINTER OUTPUT TO FILE XOUT2
[ECHO,SYSTEM]
```

Leitet alle Druckerausgaben in die Datei XOUT2 und wiederholt sie auf dem Bildschirm (mit ECHO–Option). Diese Umleitung ist so lange wirksam, bis der Befehl PUT PRINTER TO PRINTER eingegeben wird.

```
A>PUT CONSOLE OUTPUT TO CONSOLE
```

Leitet die Ausgabe wieder auf die Konsole.

```
A>PUT PRINTER OUTPUT TO PRINTER
```

Leitet die Druckerausgabe wieder auf den Drucker.

RENAME–Befehl

Format: RENAME {DATEINAME–NEU = DATEINAME–ALT}

Zweck: RENAME ermöglicht das Umbenennen von Diskettendateien. Mehrere Namen können gleichzeitig bei Verwendung der Zeichen * oder ? geändert werden. Der Befehl kann auch mit REN abgekürzt werden.

Beispiele:

```
A>RENAME NEWFILE.BAS=OLDFILE.BAS
```

Die Datei OLDFILE.BAS wird in NEWFILE.BAS umbenannt.

```
A>RENAME
```

Es erscheinen folgende System–Abfragen:

```
Enter New Name: Y.PRN
Enter Old Name: X.PRN
Y       .PRN=X          .PRN
A>
```

```
B>REN A:PRINTS.NEW=PRINCE.NEW
```

Die Datei PRINCE.NEW im Laufwerk A wird in PRINTS.NEW umbenannt.

```
A>RENAME S*.TEX=A*.TEX
```

Alle Dateien, auf die die Abkürzung A*.TEX zutrifft, werden in S*.TEX unbenannt.

```
A>REN B:NEWLIST=B:OLDLIST
```

Die Datei OLDLIST wird in NEWLIST umbe– nannt. Die Laufwerksbezeichnung muß nur einmal angegeben werden. Die nachfolgenden Befehlszeilen haben die gleiche Auswirkung:

```
A>REN B:NEWLIST=OLDLIST
```

oder

```
A>REN NEWLIST=B:OLDLIST
```

RMAC–Befehl

Format: RMAC dateispez { $Rd / $Sd / $Pd }

Zweck: Der Assembler RMAC erzeugt einen verschiebbaren Kode in einer REL–Datei, der mit einem Linker in eine COM–Datei gewandelt werden kann.

Optionen: Die Optionen bestimmen die Zuordnung der Ausgabedateien zu den Ausgabeeinheiten. Der Buchstabe d wird dabei durch die korrekte Bezeichnung ersetzt.

Tabelle 7.22 CP/M 3.0: RMAC–Optionen

Option	Bedeutung
Rd	Einheit für die REL–Datei (A – E, Z)
Sd	Einheit für die SYM–Datei (A – E, X, P, Z)
Pd	Einheit für die PRN–Datei (A – E, X, P, Z)

Hierbei bedeutet:

A – E	Laufwerke A bis E
X	Konsole als Ausgabeeinheit
P	Drucker als Ausgabeeinheit
Z	unterbindet die Ausgabe

Beispiel:

```
A>RMAC TEST $PX $B B
```

Die Datei TEST wird vom Laufwerk A assembliert, die Ausgabedatei TEST.PRN auf der Konsole, die Symboldatei TEST.SYM und die verschiebbare Objektdatei TEST.REL auf Laufwerk B ausgegeben.

SAVE–Befehl

Format: SAVE

Zweck: SAVE kopiert den Inhalt eines Teils des Speichers in eine Datei. Bei Verwendung des SAVE–Befehls wird dieser zuerst eingegeben und dann ein Programm (z. B. SID), das eine Datei in den Speicher einliest, gestartet. SAVE meldet sich dann mit einigen Abfragen wieder.

Beispiel:

```
A>SAVE
A>SID DUMP.COM
```

Das eingelesene Programm wird gestartet:

```
*g0
```

oder mit ↑C abgebrochen. Wenn dieses Programm beendet oder abgebrochen wird, unterbricht SAVE die Rückkehr in das System und meldet sich wie folgt:

```
SAVE Ver 3.0
Enter file (type RETURN to exit): dump2.com
```

Wenn die Datei DUMP2.COM bereits existiert, wird gefragt:

```
Delete dump2.com? Y
```

Dann fragt das System nach dem Speicherbereich, der abgespeichert werden soll:

```
Beginning hex address 100
Ending hex address 400
```

Der Speicherinhalt von 100H bis 400H wird in die Datei DUMP2.COM kopiert.

SET–Befehl

Format: SET [Optionen]
SET d: [Optionen]
SET dateispez [Optionen]

Zweck: Mit dem SET–Befehl werden Dateiattribute (Read Write, Read Only, DIR und SYS), der Paßwort–Schutz und die Art der Zeit– und Datumskennung bestimmt. Um die Zeitkennung der Dateien zu initialisieren, muß zuerst INITDIR ausgeführt werden.

Datei–Attribute setzen

Format: SET dateispez <Optionen>

Tabelle 7.23 CP/M 3.0: Datei–Attribute

Attribut	Bedeutung
DIR	Systemattribut der Datei wird aufgehoben.
SYS	Datei wird als Systemdatei (SYS) gekennzeichnet
RO	Datei wird als Read–Only gekennzeichnet
RW	Datei wird als Read–Write gekennzeichnet

Tabelle 7.23 CP/M 3.0: Datei–Attribute (Fortsetzung)

Attribut	Bedeutung
ARCHIVE = OFF	Das Archivattribut wird entfernt. Das bedeutet, daß die Datei beim Kopieren nicht mitkopiert wird. Siehe unter PIP mit Option [A].
ARCHIVE = ON	Das Archivattribut wird auf EIN gesetzt.
F1 = ON / OFF	Setzen oder Auschalten des vom Benutzer definierbaren Attributs F1.
F2 = ON / OFF	Setzen oder Auschalten des vom Benutzer definierbaren Attributs F2.
F3 = ON / OFF	Setzen oder Auschalten des vom Benutzer definierbaren Attributs F3.
F4 = ON / OFF	Setzen oder Auschalten des vom Benutzer definierbaren Attributs F4.

Beispiele:

```
A>SET MYFILE.TXT [RO,SYS]
```

Der Datei MYFILE.TXT werden die Nur–Lese– (RO) und SYS–Attribute zugeordnet.

```
A>SET DEINFILE.TXT <RW,DIR>
```

Der Datei DEINFILE.TXT werden die Schreib– Lese– (RW) und DIR–Attribute zugeordnet.

Laufwerks–Attribut bestimmen

Format: SET {d:} [RO]
SET {d:} [RW]

Zweck: Den Laufwerken kann mit dem SET–Befehl entweder das Nur–Lese– (RO) oder das Schreib– Lese–Attribut zugeordnet werden.

Ein Laufwerk, dem das Nur–Lese–Attribut zugeordnet ist, kann nur zum Lesen von Dateien verwendet werden. Der PIP–Befehl kann keine Dateien auf Disketten in diesem Laufwerk kopieren, mit ERASE können keine Dateien gelöscht und mit RENAME keine Dateien umbenannt werden.

Beim Eingeben von CTRL–C wird allen Lauf– werken das Schreib–/Lese–Attribut zugeordnet (Reset der Laufwerke).

Beispiel:

```
A>SET B: [RO] <CR>
```

Das Laufwerk B wird als Nur–Lese–Laufwerk markiert.

```
A>SET C: [RW] <CR>
```

Das Laufwerk C wird als Schreib–Lese–Laufwerk (RW) markiert. Der Status von anderen Laufwerken wird nicht verändert.

```
↑C
```

Alle Laufwerke werden als Schreib–/Lese– Laufwerke (RW) gesetzt.

Disketten–Label festlegen

FORMAT: SET {d:} [NAME = LABELNAM.TYP]

Zweck: Zuordnung eines Namens (Label) für die Diskette. Für die Zusammensetzung des Labels gelten die gleichen Regeln wie für den Namen und die Typbezeichnung einer Datei.

Beispiel:

```
A>SET B: [NAME=BEISPIEL.TXT]<CR>
```

Die Diskette im Laufwerk B bekommt das Label BEISPIEL.TXT zugeordnet.

Disketten–Label ein Paßwort zuordnen

Format: SET [PASSWORD = passwort]
SET [PASSWORT = < CR >

Bei Disketten, die mit einem durch ein Paßwort geschützten Label versehen sind, können das Label oder das Paßwort nur nach Eingabe des korrekten bisherigen Paßworts geändert werden. Das SET–Programm fragt dann selbständig nach dem alten Paßwort.

SET [PASSWORD = passwort]
Dem Disketten–Label wird ein Paßwort zugeordnet.

A > SET [PASSWORD = < CR >
Ohne die Eingabe einer Bezeichnung nach dem Gleichheitszeichen wird das existierende Paßwort gelöscht.

Paßwort–Schutz für Diskettendateien ein– oder ausschalten

Format: SET [PROTECT = ON]
SET [PROTECT = OFF]

Zweck: Mit PROTECT = ON wird ein Paßwortschutz für alle Dateien auf der Diskette gesetzt. Dieser Befehl muß ausgeführt werden, bevor einzelnen Dateien auf der Diskette eigene Paßworte zugeordnet werden. Mit PROTECT–OFF wird diese Option wieder ausgeschaltet. Mit dem SHOW–Befehl kann jederzeit überprüft werden, ob die Diskette mit einem Paßwort geschützt ist oder nicht.

Paßwortschutz Diskettendateien zuordnen

Format: SET dateispez [PASSWORD = passwort]

Zweck: Diese Art des SET–Befehls ordnet den mit 'dateispez' definierten Dateien einen Paßwortschutz zu. Das Paßwort kann bis zu acht Zeichen lang sein. Kleingeschriebene Zeichen werden in großgeschriebene Zeichen umgewandelt.

Bei der Dateienbestimmung können auch die Platzhalterzeichen * und ? verwendet werden. SET ordnet dann dasselbe Paßwort allen Dateien zu, auf die die Dateibezeichnung zutrifft.

Bemerkung: Notieren Sie sich immer die von Ihnen festgelegten Paßworte für Ihre Dateien. Ohne das Paßwort zu kennen, ist es nicht möglich, auf die geschützten Dateien zuzugreifen. Wenn Sie das Paßwort für das Diskettenlabel vergessen, können Sie den Paßwortschutz für die Diskette nicht wieder ausschalten.

Beispiel:

```
>SET MEINDAT.TXT [PASSWORD=SCHUTZ]
```

SCHUTZ ist das Paßwort, das der Datei MEINDAT.TXT zugewiesen wird.

Art des Paßwortschutzes für geschützte Dateien setzen

Format:

SET dateispez [PROTECT=READ]
SET dateispez [PROTECT=WRITE]
SET dateispez [PROTECT=DELETE]
SET dateispez [PROTECT=NONE]

Zweck:

Der Paßwortschutz von Dateien kann in vier verschiedenen Stufen stattfinden. Die Schutzmodi sind READ, WRITE, DELETE und NONE. Die Bedeutungen dieser Modi sind in der Tabelle 7.23 erklärt.

Beispiele:

```
B>SET *.TXT [PASSWORD=NOS, PROTECT=WRITE]
```

Das Paßwort NOS wird allen Dateien des Typs TXT auf der Diskette im Laufwerk B zugeordnet. Jede dieser Dateien ist gegen das unberechtigte Schreiben und Verändern geschützt. Diesen Schutz kann nur derjenige Benutzer aufheben, der das Paßwort kennt.

Tabelle 7.24 CP/M 3.0: SET–Passwort–Optionen

Option	Schutzart
READ	Paßwortangabe ist zum Lesen, Kopieren, Schreiben, Löschen oder Umbenennen notwendig.
WRITE	Paßwortangabe ist zum Schreiben, Löschen oder Umbenennen notwendig. Zum Lesen ist kein Paßwort erforderlich.
DELETE	Paßwortangabe ist nur zum Löschen oder Umbenennen der Datei notwendig. Zum Lesen oder Modifizieren der Datei ist kein Paßwort erforderlich.
NONE	Der Datei wird kein Paßwort zugeordnet. Bei existierendem Paßwort kann mit dieser Option das Paßwort gelöscht werden.

Standard–Paßwort (default) bestimmen

Format: SET [DEFAULT = passwort]

Zweck: Mit dem SET–Befehl für das Standard–Paßwort wird für die Einschaltdauer des Computers das vorgegebene Paßwort als Standard–Paßwort verwendet. Auf alle Dateien, die mit demselben Paßwort wie das Standard–Paßwort geschützt sind, kann zugegriffen werden.

Wird bei der Dateierstellung kein Paßwort bestimmt und die PROTECT–Option ist auf ON gesetzt, wird diesen Dateien das Standard–Paßwort als ihr Paßwort zugeordnet.

Beispiel:

```
A>SET [DEFAULT=MUSTER]
```

Das System setzt MUSTER als Ersatz–Paßwort ein, wenn kein anderes Paßwort für die Datei angegeben wird.

Zeit– und Datumskennungsart

Format:

SET [CREATE = ON]
SET [ACCESS = ON]
SET [UPDATE = ON]

Zweck:

Mit diesen SET–Optionen wird ermöglicht, Zeit– und Datumsmarkierungen über Zugriffe auf die Diskettendateien zu speichern.

[CREATE = ON]
Nur wenn diese Option eingeschaltet ist, werden den Dateien Zeitmarkierungen über den Zeitpunkt (Datum und Zeit) der Erstellung im Inhaltsverzeichnis der Diskette zugeordnet.

[ACCESS = ON]
Im Inhaltsverzeichnis der Diskette wird der Zeitpunkt des letzten Zugriffs auf die jeweilige Datei festgehalten, wenn der Datei das Attribut [RW] zugeordnet ist.

CREATE = ON und ACCESS = ON schalten sich gegenseitig aus.

[UPDATE = ON]
schaltet die Erstellung der Zeitmarkierungen über die letzte Dateiveränderung ein.

Beispiel:

```
A>SET [ACCESS=ON]
```

Nach Eingabe dieses Befehls wird mit dem Befehl **DIR** [**FULL**] z. B. folgendes angezeigt:

```
B>DIR [FULL]

 Name       Bytes Recs  Attributes Prot    Update          Access

ONE     .TEX  9k   71    Dir RW     None                  08/03/81 10:56
THREE   .TEX  12k  95    Dir RW     None                  08/05/81 15:45
TWO     .TEX  10k  76    Dir RW     None                  08/10/81 09:13
```

In der Spalte 'Access' wird jeweils der Zeitpunkt des letzten Zugriffs auf die jeweilige Datei angezeigt. Das Anzeigen des Dateinamens durch DIR wird nicht als Dateizugriff bewertet.

Nach der Eingabe von

```
A>SET [CREATE=ON,UPDATE ON]
```

wird mit dem Befehl **DIR** [**FULL**] z. B. folgendes angezeigt:

```
A>DIR [FULL]
```

Name	Bytes	Recs	Attribute	Prot	Update	Access
BEISPIEL.TEX	109k	873	Dir RW	None	08/05/81 14:01	08/01/81 09:36
TEXTE.TEX	59k	475	Dir RW	None	08/08/81 12:11	08/01/81 09:40
TWO.TEX	76k	76	Dir RW	None	08/08/81 08:46	08/01/81 10:15

SETDEF–Befehl

Format:

SETDEF {d:{,d:{,d:{d:}}}} {[TEMPORARY = d:]
[ORDER = (typ {,typ})]}

SETDEF [DISPLAY / NO DISPLAY]

SETDEF [PAGE / NO PAGE]

Zweck:

Der SETDEF–Befehl wird dazu benutzt, bestehende Diskettendateien– und Dateityp–Suchkriterien und die Laufwerksbezeichnungen für die Ablage von Zwischendateien zu bestimmen oder bestehende Zuordnungen anzuzeigen.

Mit SETDEF können auch die DISPLAY– und PAGE–Modi ein– und ausgeschaltet werden.

Wenn der DISPLAY–Modus eingeschaltet ist, [DISPLAY = ON], werden die Namen und Typbezeichnungen der jeweils geladenen oder von SUBMIT ausgeführten Dateien auf dem Bildschirm angezeigt. Bei [DISPLAY = OFF] wird diese Anzeige unterdrückt.

Beim eingeschalteten PAGE–Modus wird die Bildschirmanzeige nach jeder voll ausgefüllten Bildschirmseite angehalten, um sie lesen zu können.

CP/M sucht gewöhnlich nur auf dem aktuellen Laufwerk nach Dateien, wenn kein anderes Laufwerk angegeben wird. Mit SETDEF kann udiese Suche auch auf andere Laufwerke erweitert werden, allerdings nur zum Ausführen von COM– und SUB–Dateien.

Anzeige der bestehenden Systemeinstellungen

Format: SETDEF

Beispiel:

```
A>SETDEF

Drive Search Path:
1st Drive              - Default
Search Order           - COM
Temporary Drive        - Default
Console Page Mode      - On
Program Name Display   - Off
```

Die oben aufgeführte Anzeige erscheint, wenn der SETDEF–Befehl gleich nach dem Systemstart eingegeben wird und noch keine Veränderungen veranlaßt wurden.

Zuordnung des Laufwerks für Zwischendateien

Beispiel:

```
A>SETDEF <TEMPORARY=D:>
```

Alle Zwischendateien werden nicht auf dem aktuellen, sondern auf dem mit dem SETDEF– Befehl angegebenen Laufwerk abgelegt.

Dateisuchkriterien bestimmen

Format: SETDEF [d:[,d:[,d:[,d:]]]]

Zweck: Der Suchweg für COM– oder SUB– Dateien wird bestimmt. Wird ein Stern (*) im Suchpfad eingegeben, wird anschließend auch auf dem jeweils aktuellen Laufwerk gesucht.

Beispiel:

```
A>SETDEF C:,*
```

Nach einer Datei wird zuerst auf Laufwerk C und anschließend, falls die Datei nicht gefunden wurde, auf dem aktuellen Laufwerk gesucht.

```
A>SETDEF B:,E:,*
```

Gesucht wird in folgender Reihenfolge: Laufwerk B, dann virtuelles Laufwerk E und anschließend das aktuelle Laufwerk.

Bestimmen der Dateitypen für das Suchkriterium

Format: SETDEF [ORDER = (typ [,typ])]

Zweck: Als Dateityp kann nur COM oder SUB eingesetzt werden. CP/M setzt nach dem Start nur COM als Vorgabe ein.

Beispiele:

```
A>SETDEF [ORDER=(SUB,COM)]
```

Gesucht wird nach SUB– und COM– Dateien.

```
B>SETDEF [ORDER=(SUB)]
```

Gesucht wird nur nach SUB–Dateien.

System–Anzeigemodus setzen

Format: SETDEF [DISPLAY / NO DISPLAY]

Beispiele: A > SETDEF [DISPLAY]

Die Anzeige von Namen und Typen der geladenen Dateien wird eingeschaltet.

Wird nach dem Einschalten der Display– Option z. B. PIP gestartet, erscheint:

```
>PIP
A:PIP    COM
CP/M 3 PIP VERSION 3.0
*
```

auf dem Bildschirm. Ist ein anderer Benutzerbereich als 0 eingeschaltet und befindet sich PIP nicht in diesem Benutzerbereich, sondern im Bereich 0 als eine System–Datei, dann wird mit:

```
4A>PIP
A:PIP    COM (User 0)
CP/M 3 PIP VERSION 3.0
*
```

angezeigt, daß PIP aus dem Benutzerbe–reich 0 geladen wurde.

Nach dem CP/M–Start ist der Display–Modus ausgeschaltet.

System–PAGE–Modus ein– oder ausschalten

Format: SETDEF [PAGE / NO PAGE]

A > SETDEF [NO PAGE]

Die Bildschirmausgabe erfolgt ohne Unterbrechung, auch wenn der angezeigte Text größer als die Bildschirmlänge ist.

Nach dem CP/M–Start ist der PAGE–Modus eingeschaltet.

SHOW–Befehl

Format: SHOW{d:} {[SPACE,LABEL,USERS,DIR,DRIVE]}

Zweck: Mit dem SHOW–Befehl können folgende Informationen über Disketten und Laufwerke angezeigt werden:

- Zugriffsstatus und freier Platz auf Diskette
- Disketten–Label
- Aktueller Benutzerbereich
- Anzahl der Dateien in einzelnen Benutzerbereichen auf der Diskette.
- Anzahl der freien Inhaltsverzeichnis– (directory–) Einträge auf der Diskette
- Laufwerksparameter

Zugriffsstatus und freien Platz auf der Diskette anzeigen

Format: SHOW {d:}{[SPACE]}

Zweck: Angezeigt wird die Laufwerksbezeichnung, der Zugriffsstatus und der freie Platz auf der Diskette für alle angemeldeten Laufwerke.

Beispiele:

```
A>SHOW B:
B: RW, Space              4k

A>SHOW
A: RW, Space            132k
B: RW, Space              4k
```

Diskettenlabel anzeigen

Format: SHOW {d:} [LABEL]

Zweck: Angezeigt werden alle Informationen über das Diskettenlabel und seine Attribute

Beispiel:

```
A>SHOW B: [LABEL]

Label for Drive B:

Directory      Passwds  Stamp    Stamp
Label          Reqd     Access   Update   Label Created    Label Updated

DISKLAB.ONE      on       on       on     06/12/85 14:06  06/13/85 22:05
```

In der ersten Spalte wird das Label der Diskette im Laufwerk B, in den nächsten Spalten, ob ein Paßwortschutz und die Zeit– und Datumsmarkierungen eingeschaltet sind, angezeigt. Anschließend werden die Zeitpunkte der Labelerstellung und der letzten Veränderung dargestellt.

Benutzerbereiche auf der Diskette anzeigen

Format: SHOW [d:] [USERS]

Zweck: Die Nummern des aktiven Benutzerbereichs und aller vorhandenen Benutzerbereiche auf der Diskette werden angezeigt, dazu die Anzahl der Dateien in den einzelnen Benutzerbereichen.

```
A>SHOW [USERS]

A: Active User  :      0
A: Active Files :      0    2    3
A: # of files   :      24   12   6

A: Number of time/date directory entries: 32
A: Number of free directory entries:      50
```

Die Anzeige der Zeit- und Datumseinträge geschieht nur bei Disketten, die mit INITDIR bearbeitet wurden.

Anzeige freier Einträge im Diskettenverzeichnis

SHOW {d:} [DIR]

A > SHOW C: [DIR]

A:Number of free directory entries: 50

Anzeige der Disketten–Eigenschaften

Format: SHOW [DRIVE]

Beispiele:

```
A>SHOW [DRIVE]

     A: Drive Characteristics
 2,720: 128 Byte Record Capacity
   340: Kilobyte Drive Capacity
   128: 32 Byte Directory Entries
   128: Checked Directory Entries
   256: Records / Directory Entry
    16: Records / Block
     8: Sectors / Track
     0: Reserved Tracks
  1024: Bytes / Physical Record

     A: >SHOW B: [DRIVE] ; im Laufwerk B ist
                           eine Kaypro-Diskette

     A: Drive Characteristics
 3,152: 128 Byte Record Capacity
   394: Kilobyte Drive Capacity
   128: 32 Byte Directory Entries
   128: Checked Directory Entries
   256: Records / Directory Entry
    16: Records / Block
    40: Sectors / Track
     1: Reserved Tracks
   512: Bytes / Physical Record
```

SID–Befehl

Format: SID {pgm–dateispez} {,sym–dateispez}

Zweck: Mit dem SID (Symbolic Instruction Debugger) kann man Programme, die für den Prozessor 8080 entwickelt sind, anzeigen und austesten. SID unterstützt Echtzeit–Unterbrechungspunkte, schrittweise Programmabarbeitung mit Ablaufanzeige, symbolische Dissassemblierung, Assemblierung, Anzeige und Veränderungen von Speicherinhalten.

SID ohne Angabe einer Dateibezeichnung lädt nur sich selbst in den Speicher. Dann können Speicherinhalte angezeigt oder einfache Programme mit dem A–Befehl geschrieben werden. Ohne ein ablauffähiges Programm im Speicher können die SID–Befehle G, T oder U nicht ausgeführt werden.

SID mit Angabe einer Dateibezeichnung lädt beides, sich selbst und die angegebene Datei, in den Speicher. Wird nur der Dateiname ohne Typbezeichnung angegeben, nimmt SID den Typ COM als Vorgabe. Mit den Befehlen C, G, T oder U kann das geladene Programm unter Kontrolle vom SID abgearbeitet werden.

Mit CTRL–S wird die Bildschirmausgabe angehalten, mit CTRL–Q wieder gestartet. Abgebrochen wird die Ausgabe durch das Drücken einer beliebigen Taste.

Mit CTRL–C wird der SID–Programmablauf abgebrochen und zurück in CP/M gesprungen.

Ausführliche SID–Beschreibung finden Sie im SID-Handbuch von Digital Research.

Tabelle 7.25 CP/M 3.0: SID–Befehle

Befehl	Format	Bedeutung
Assemble	As	Assembler–Befehl eingeben, s ist die Startadresse.
Call	Cs {b{,d}}	Sprung auf die Adresse s, b ist der Wert im Registerpaar BC und d der Wert im Registerpaar DE.
Display	D{W}{s}{,f}	Speicherinhalt in HEX und ASCII anzeigen. Mit W kann das 16–Bit–Wortformat angegeben werden, s ist die Startadresse und f die Endadresse.
Load	Edateispez {,sym–dateispez}	Ein Programm und Symboltabelle zuladen.
Load	E* sym–dateispez	Symbolische Tabelle zuladen.
Fill	Fs,f,d	Speicher mit einem Wert füllen, s ist die Startadresse, f die Endadresse und d ein 8–Bit–Wert zum Auffüllen.
Go	G{p}{,a{,b}}	Programmausführung starten. p ist die Startadresse, a ist der erste und b der zweite Haltepunkt. Nach der Programmausführung erfolgt ein Warmstart und Sprung in CP/M.
Hex	Ha	Dezimal– und ASCII–Wert der Hexzahl a anzeigen.
	H#b	Hex–Wert der Dezimalzahl b anzeigen.
	Ha,b	Die Differenz und Summe der zwei Hexzahlen a und b berechnen und anzeigen.
Input	Ibefehl	CCP–Befehlszeile eingeben.

Tabelle 7.25 CP/M 3.0: SID–Befehle (Fortsetzung)

Befehl	Format	Bedeutung
List	L{s}{,f}	8080–Befehle mnemonisch auflisten. s ist die Start– und f die Endadresse.
Move	Ms,h,d	Speicherinhalte zwischen der Startadresse s und Endadresse f ab Adresse d kopieren.
Pass	P{p{,c}}	Pass–Unterbechungs–Punkt setzen, entfernen oder anzeigen. p ist die Unterbrechungsadresse und c der Startwert des Ablaufzählers.
Read	Rdateispez{,d}	Mit diesem Befehl wird eine Datei in den Speicher geladen. Wahlweise kann ein Versatz d eingegeben werden.
Set	S{W}s	Werte in den Speicher eingeben. Bei Eingabe von W wird ein 16–Bit–Wort erwartet. Eine ASCII–String–Eingabe kann nach einem '"' erfolgen.
Trace	T{n{,c}}	Programmabarbeitung in Einzelschritten. n ist die Anzahl der Schritte und c ist die Adresse für Trace–Dienstprogramme vom Typ UTL.
Trace	T{W}{n{,c}}	Bei Eingabe von W werden Unterprogramme schnell und nicht schrittweise abgearbeitet.
Untrace	U{W}{n{,c}}	Kontrollierte, jedoch nicht schrittweise Abarbeitung. Parametereingabe wie bei T.

Tabelle 7.25 CP/M 3.0: SID–Befehle (Fortsetzung)

Befehl	Format	Bedeutung
Value	V	Anzeige der nächsten verfügbaren Adresse (NEXT), der End + 1–Adresse der längsten zugeladenen Datei im 'MSZE', der aktuellen Adresse im Programmschrittzähler (PC) und der Endadresse des verfügbaren Speichers (END).
Write	Wdateispez{,s,f}	Schreiben der Speicherinhalte in die angegebene Datei von der Adresse s bis Adresse f.
Examine	X{f} X{r}	Ansehen oder Ändern der CPU–Register. f ist eines der Statusregister-Bits C, E, I, M oder Z; r ist ein Register A, B, D, H, P oder S.

Beispiele:

```
A>SID
```

SID wird in den Speicher geladen und gestartet. Seine Bereitschaft zeigt das Programm mit dem Zeichen #.

```
A> BEISPIEL .HEX
```

Nach dem Laden werden folgende Speicheradressen angezeigt:

```
NEXT MSZE PC END nnnn mmmm pppp eeee
```

Hierbei ist:

nnnn eine hexadezimale Angabe für die nächste freie Adresse nach dem zugeladenen Programm,

mmmm die nächste freie Adresse nach dem längsten bisher zugeladenem Programm (nach dem SID–Aufruf ist nnnn = mmmm),

pppp die Adresse im Programmzähler, ab welcher die Programmabarbeitung mit G ohne Zusatzangaben (in der Regel 0100H) beginnt und

eeee die Endadresse des verfügbaren Benutzerspeichers (TPA).

```
#DFE00+#128
```

Das 'Doppelkreuz–Zeichen' # ist die SID–Bereitschaftsmeldung. D ist der Befehl zur Anzeige von Speicherinhalten und FE00 + # 128 die Angabe der Startadresse FE80h. Ohne Eingabe einer Endadresse werden 12 Zeilen ausgegeben.

SUBMIT–Befehl

Format: SUBMIT {dateispez} {Argument}. . .{Argument}

Zweck: Mit dem SUBMIT–Befehl wird eine Datei des Typs SUB aufgerufen und die darin enthaltenen CP/M–Befehle ausgeführt. Mit einem einzigen Befehl wird eine beliebige Anzahl von CP/M–Befehlen und –Programmen im Stapelmodus (batch mode) ausgeführt.

Die in der SUB–Datei enthaltenen Befehle werden so ausgeführt, als ob sie über die Tastatur eingegeben wären.

Zur Verkettung von Dateien, d. h. zur aufeinanderfolgenden Ausführung mehrerer SUB–Dateien, kann mit der SUB–Datei wieder eine SUB–Datei aufgerufen werden.

Argumente in der Befehlszeile können an die von SUBMIT aufgerufenen Programme über die Übergabeparameter $1 bis $9 übergeben werden. Das bedeutet, daß alle Parameter, die nach dem SUBMIT–Aufruf in der Befehlszeile eingegeben sind, in der eingegebenen Reihenfolge den Übergabeparameter $1 bis $9 zugeordnet werden.

Wenn Sie die folgende Datei START.SUB

```
ERA $1.BAK
DIR $1
PIP $1=A:$2.COM
```

erstellen und SUBMIT mit folgender Eingabe in der Befehlszeile aufrufen:

```
>SUBMIT START SAM TEX
```

dann wird jedesmal anstelle $1 das Argument SAM und anstelle $2 TEX verwendet. Dazu erstellt SUBMIT eine

$$$.SUB–Zwischendatei, in der die korrekten Parameter eingesetzt sind:

```
ERA SAM.BAK
DIR SAM
PIP SAM=A:TEX.COM
```

Erst diese Zwischendatei wird von SUBMIT als die endgültige Befehlsdatei verwendet.

Programmeingaben bei SUBMIT–Ausführung

In einer SUB–Datei können auch Eingaben für die aufgerufenen Programme vorgegeben werden. Jede Eingabe wird mit '<' (spitze Klammer auf) eingeleitet, wie in folgender Beispielsdatei:

```
PIP
<B:=*.ASM
<CON:=DUMP.ASM
<
DIR
```

Die drei Zeilen nach dem PIP–Aufruf sind Eingaben für den PIP–Befehl. Die '<' in der dritten Zeile bedeutet <CR>. Damit wird PIP–Ablauf beendet und in CP/M zur Ausführung des nächsten Befehls, DIR, gesprungen.

Wenn das ablaufende Programm weniger Eingaben als in der SUB–Datei vorgesehen verlangt, werden die übrigen Eingaben ignoriert und SUBMIT gibt folgende Meldung aus:

```
Warning: Programm input ignored
```

Werden von dem ablaufenden Programm mehr Eingaben, als in der SUB–Datei enthalten, verlangt, werden diese Eingaben von der Tastatur erwartet.

Die SUB–Datei

In der SUB–Datei können folgende Daten verwendet werden:

- Jeder gültige CP/M–Befehl
- Jeder gültige CP/M–Befehl mit SUBMIT Argumenten
- Entsprechende Programmeingabezeilen
- Eingabezeilen mit Parameter $1 – $9

Auch hier dürfen die Zeilen höchstens 128 Zeichen lang sein.

Beispiel:

```
DIR
DIR*.BAK
MAC $1 $$$4
PIP LST:=$1.PRN[T$2 $3 $5]
DIR*.ASM
PIP
<B:=*.ASM
<CON:=DUMP.ASM
<
DIR B:
```

SUBMIT–Befehl ausführen

Format: SUBMIT
SUBMIT dateispez
SUBMIT dateispez argument . . .argument

Zweck: Wird nur SUBMIT ohne weitere Angaben eingegeben, fragt das System nach den restlichen Eingaben. Dann können noch Dateibezeichnung und Argumente eingegeben werden.

Beispiele:

```
>A:SUBMIT<CR>
```

Enter File to Submit: START B TEX

```
>A:SUBMIT SUBA<CR>
```

Die SUB–Datei SUBA wird ausgeführt.

```
>A:SUBMIT AA ZZ SZ<CR>
```

AA bedeutet die SUB–Datei AA.SUB,
ZZ wird anstelle aller $1– und
SZ anstelle aller $2–Übergabeparameter in der AA.SUB–Datei verwendet.

Autostartdatei PROFILE.SUB

Jedesmal, wenn CP/M gestartet wird, sucht das System nach einer SUB–Datei mit dem Namen PROFILE.SUB und führt sie, wenn sie vorhanden ist, selbstätig aus. Damit kann automatisch jeder beliebige Perogrammablauf ausgeführt werden. Beispielsweise kann diese Datei folgenden Text beinhalten:

```
DATE SET
```

Dann wird nach jedem Systemstart automatisch der Befehl DATE im Dialog ausgeführt.

TYPE–Befehl

Format: TYPE {dateispez {[PAGE]}|[NO PAGE]}

Zweck: Der TYPE–Befehl zeigt den Inhalt von ASCII– (Text–)Dateien auf dem Bildschirm an. Die PAGE–Option stoppt die Ausgabe jedesmal nach Darstellung von n–Zeilen. Diese Zeilenzahl entspricht gewöhnlich der Systemvoreinstellung auf 24 Zeilen und kann mit dem DEVICE–Befehl geändert werden. TYPE–Ablauf kann mit CTRL–C unterbrochen werden.

Wird NO PAGE als Argument eingegeben, geschieht die Ausgabe ohne Unterbrechung.

Wird nur TYPE ohne eine Dateispezifikation eingegeben, erscheint folgende Aufforderung:

Enter filename:

und der Dateiname kann eingegeben werden.

TAB–Zeichen werden bei der Ausgabe auf8 Spalten expandiert. Mit CTRL–S kann die Ausgabe jederzeit angehalten und mit CTRL–Q wieder fortgesetzt werden.

Mit CTRL–P kann die Ausgabe gleichzeitig auf dem Bildschirm und einem Drucker erfolgen.

Beispiele:

```
>A:TYPE MEINPROG.TXT<CR>
```

Die Ausgabe des Inhalts der Datei MEINPROG.TXT wird jedesmal nach 24 Zeilen angehalten.

```
>A:TYPE B:NEUDATEI.BEI <NO PAGE><CR>
```

Die Ausgabe der Datei NEUDATEI.BEI erfolgt kontinuierlich von Laufwerk B.

Bemerkung: Wird mit dem TYPE–Befehl versucht, den Inhalt einer Binären–(nicht–ASCII)–Datei auszugeben, kann es zu den wildesten Effekten auf dem Bildschirm kommen, da diese Dateien auch die nichtdruckbaren Zeichen beinhalten.

Der Inhalt solcher Dateien kann mit dem DUMP–Befehl ausgegeben werden.

USER–Befehl

Format: USER {nummer}

Mit dem USER–Befehl wird der aktuelle Benutzerbereich gesetzt. Nach dem CP/M–Start ist der Benutzerbereich 0 voreingestellt. Mit dem USER–Befehl kann einer der Benutzerbereiche 0 bis 15 als der aktuelle Benutzerbereich zugeordnet werden.

CP/M 3.0 ordnet jeder Datei die zugehörige Benutzerbereichsnummer zu. Gewöhnlich kann man nur auf Dateien zugreifen, die in dem aktuellen Benutzerbereich liegen. Eine Ausnahme bilden die Dateien im Benutzerbereich 0, die das SYS–Attribut zugeordnet haben. Auf diese Dateien kann auch aus jedem anderen Benutzerbereich zugegriffen werden.

Beispiele:

```
A>USER
Enter User#:5
5A>
```

Nach der Eingabe von 5 wird der Benutzerbe– reich 5 als der aktuelle Bereich zugeord– net.

```
A>USER 3
3A>
```

Mit diesem Befehl wird in den Benutzerbe– reich 3 gewechselt.

XREF–Befehl

Format: XREF {d:} dateispez {$P}

Zweck: Mit dem Befehl XREF wird eine Referenztabelle der Variablen des mit 'dateispez' angegebenen Programms erstellt. XREF benötigt dazu die entsprechenden PRN– und SYM–Dateien, die von MAC oder RMAC erstellt wurden.

Die SYM– und PRN–Datei müssen den im 'dateispez' angegebenen Dateinamen haben. Die Ausgabedatei bekommt die Typbezeichnung XRF.

Beispiele:

```
A>XREF B:MEINPROG
```

In diesem Beispiel wird XREF von Laufwerk A aufgerufen. Verarbeitet werden die Dateien MEINPROG.PRN und MEINPROG.SYM, die sich auf der Diskette im Laufwerk B befinden müssen. Die erzeugte Ausgabedatei MEINPROG.XRF wird auf die Diskette im Laufwerk B geschrieben.

```
A>XREF B:MEINPROG $P
```

Mit der Eingabe der Option $P erfolgt die Ausgabe direkt auf einem Drucker.

Commodore Büromaschinen GmbH,
Lyoner Str. 38, 6000 Frankfurt 71
Tel. (069) 6638-0, Telefax 6638-159
Telex 4 185 663 como d

Commodore AG,
Langenhagstraße 1, CH-4147 Aesch
Tel. (061) 78 22 12, Twx. 64 961

Commodore Büromaschinen GmbH,
Kinskygasse 40-44, A-1232 Wien,
Tel. (02 22) 67 56 00, Twx. 111 350

Commodore

Art.-Nr.: 580128